Zentrum für sichere Informationstechnologie – Austria
Secure Information Technology Center – Austria

A-1030 Wien, Seidlgasse 22 / 9
Tel.: (+43 1) 503 19 63–0
Fax: (+43 1) 503 19 63–66

A-8010 Graz, Inffeldgasse 16a
Tel.: (+43 316) 873-5514
Fax: (+43 316) 873-5520

http://www.a-sit.at
E-Mail: office@a-sit.at

DVR: 1035461
ZVR: 948166612
UID: ATU60778947

SICHERHEITSANALYSE MOBILE PLATTFORMEN

VERSION 1.3.4 – 24.05.2016

Alexander Marsalek - alexander.marsalek@a-sit.at
Peter Teufl - peter.teufl@a-sit.at
Thomas Zefferer - thomas.zefferer@a-sit.at
Sandra Kreuzhuber

Zusammenfassung: Durch die Weiterentwicklungen bestehender- und Einführung neuer Smartphone-Plattformen sind sowohl Betreiberinnen und Betreiber von IT-Systemen als auch Benutzerinnen und Benutzer gefordert, Überblick über die ständig wachsenden Funktionen zu behalten. Mittlerweile verfügen Smartphones über beinahe dieselben Möglichkeiten zur Datenverarbeitung wie herkömmliche Desktop-Computer und führen so zunehmend auch zur Verarbeitung kritischer Unternehmensdaten am mobilen Gerät. Ein Diebstahl dieser Daten oder ein Angriff auf ein mobiles Gerät bedeutet somit nicht nur eine Verletzung der Privatsphäre der Benutzerin bzw. des Benutzers, sondern kann bei Offenlegung von Unternehmensdaten einen hohen finanziellen Schaden bedeuten. Dieses Überblickspapier soll technisch versierten Leserinnen und Lesern einen Einblick über zur Verfügung stehende Sicherheitsfunktionen bieten und diese anhand der Plattformen iOS, Windows Mobile, Android (mit zusätzlichem Fokus auf Samsung KNOX, BlackBerry PRIV und Android for Work) und BlackBerry 10 analysieren. Ziel dieses Papiers ist es, eine Grundlage für die Bewertung der Eignung einer Smartphone-Plattform für den Einsatz im Unternehmen zu bieten.
Dieses Dokument wird laufend aktualisiert. Aufgrund der schnellen Weiterentwicklung der einzelnen Plattformen und der damit verbundenen Herausforderung alle Informationen aktuell zu halten, wird bei jeder Plattform eine „Statusbox“ hinzugefügt, die dem Leser Auskunft gibt welche Version der jeweiligen Plattform behandelt wird und wann die dafür relevanten Informationen eingearbeitet wurden.

Inhaltsverzeichnis

1. Executive Summary

In diesem Dokument werden die Plattformen iOS, Windows Mobile 10, BlackBerry 10, Android, Android – Samsung Knox und Android – BlackBerry PRIV analysiert.

Generell bieten im Jahr 2016 alle Smartphone-Plattformen ausgereifte Sicherheitsfunktionen in den unterschiedlichen Bereichen (z.B. Basissicherheit, Verschlüsselung, Zugriffschutz etc.) an und können prinzipiell für unterschiedliche Einsatzgebiete mit diversen Sicherheitsanforderungen eingesetzt werden. Für die Plattform Android muss allerdings berücksichtigt werden, dass es eine sehr große Heterogenität gibt, die einen starken Einfluss auf die Sicherheitsfunktionen der Systeme hat. Diese Unterschiede ergeben sich vor allem aufgrund der Möglichkeit, dass Hersteller das Android-System anhand der eigenen Anforderungen anpassen können. Teilweise werden umfassende Erweiterungen/Änderungen vorgenommen, so dass Plattformen wie Samsung Knox oder BlackBerry PRIV schon eher als eigene Plattformen zu betrachten sind – vor allem dann, wenn detaillierte Aussagen zu Sicherheitsfunktionen aufgrund hoher Sicherheitsanforderungen benötigt werden. Die Schwierigkeit in dieser heterogenen Umgebung besteht einerseits darin die mittlerweile sehr umfassenden Sicherheitsfunktionen für die jeweiligen Einsatzgebiete entsprechend zu konfigurieren, anderseits die herstellerspezifischen Eigenschaften (z.B. Updates, Hardware-basierte Verschlüsselung, Erweiterungen) korrekt zu berücksichtigen. Darüber hinaus muss vor allem beim Einsatz im Unternehmen darauf geachtet werden, welche Funktionen zum Verwalten der Geräte zur Verfügung stehen. Diese Funktionen unterscheiden sich bei den verschiedenen Plattformen, müssen aber auch im Konnex mit den verwendeten Verwaltungssystemen betrachtet werden. Ein wesentlicher Aspekt für sicherheitskritische Anwendungen ist außerdem die Updatestrategie der jeweiligen Plattform– oder Subplattform.

Im Folgenden werden die wesentlichen Erkenntnisse der einzelnen Plattformen zusammengefasst.

iOS: iOS bietet sehr gute Sicherheitsfunktionen die es erlauben auch sicherheitskritische Daten (bei korrekter Konfiguration) am Gerät abzulegen. Die Verfügbarkeit von Updates ist bei iOS-Geräten für einen – im Vergleich zu den anderen Plattformen – langen Zeitraum gegeben. Im Unternehmensbereich stehen umfassende Verwaltungsoptionen zur Verfügung, die es erlauben die Geräte je nach Sicherheitsanforderungen zu konfigurieren. iOS-Geräte bieten umfassende Cloud-Funktionen die über das System iCloud zur Verfügung gestellt werden (Backup, Synchronisation, Speichern von Daten und Passwörtern). Die Sicherheit der im iCloud-System gespeicherten Daten hängt im Wesentlichen von der Absicherung des dafür nötigen Benutzerkontos ab (Passworteigenschaften, Zwei-Faktor-Authentifizierung). Für sicherheitskritische Bereiche ist es daher essentiell im Detail darauf zu achten welche Cloud-Funktionen für die Verwendung frei gegeben werden.

Android: In den Android Versionen 5.x und 6.x wurden zahlreiche Sicherheitsfunktionen hinzugefügt, die auch einen Einsatz von Android in Bereichen mit höheren Sicherheitsanforderungen ermöglichen. Ein wesentlicher zu beachtender Punkt ist die – aufgrund der Offenheit des Systems – Diversität der verfügbaren Android-Lösungen. Eine Vielzahl von Herstellern verwenden das Basis-Android-System und modifizieren und erweitern dieses je nach den eigenen Anforderungen. Zusätzlich wird dabei auf unterschiedliche Hardware-Implementierungen gesetzt. Für die Sicherheit des Systems ist es daher schwierig generische Aussagen zu machen, da die zeitliche Verfügbarkeit von Updates, oder die Verfügbarkeit von spezifischen hardware-basierten Funktionen (z.B. Hardware-Element zur Absicherung der Schlüssel die für die Dateisystemverschlüsselung) vom jeweiligen Hersteller abhängen. Ebenso davon betroffen ist die Verfügbarkeit von Funktionen im Mobile-Device-Management Bereich. Es muss daher eine sehr wohl überlegte Entscheidung getroffen werden, welchen Hersteller man aufgrund der jeweiligen Sicherheitsanforderungen wählt. In dieser Studie werden zwei weitere Systeme in Ergänzung zur Standard-Android Plattform analysiert – Samsung KNOX und Blackberry PRIV.

Blackberry, Windows Mobile 10: Beide Systeme bieten im Wesentlichen sehr gute Sicherheitsfunktionen in allen Bereichen (Updates, MDM etc.). Für Blackberry muss beachtet

werden, dass das Unternehmen parallel zum bisherigen Betriebssystem auch auf Android setzt (Blackberry PRIV).

Sicherheitsfunktion	*iOS*	*Android*	*Samsung Knox*	*BlackBerry PRIV*	*Windows Mobile 10*	*BlackBerry 10*
Basissicherheit:						
ASLR[1]	Diese Basissicherheitsfunktionen werden aktuell von allen Plattformen unterstützt.					
DEP[2]						
Sandboxing						
Verifizierung der geladenen Softwarekomponenten während des Bootprozesses	Ja	Ja, ab Version 4.4 (die Implementierung hängt aber vom Hersteller ab und muss für Detailfragen im Bereich Sicherheit berücksichtigt werden)	Ja	Ja	Ja	Ja
Verschlüsselung:						
Dateisystemverschlüsselung	Ja	Ja (standardmäßig aktiv seit Android 5)			Ja	Ja
Applikationsspezifisches Verschlüsselungssystem	Ja	Aktuell noch nicht in der Detailausprägung wie bei iOS (siehe Protection Class), es steht aber die Android KeyChain zur Verfügung die dem Entwickler zur Verfügung steht, um spezielle Anforderungen abdecken zu können.			Ja (noch nicht in der Detailausprägung von iOS – siehe Protection Class)	n/a
Verschlüsselungssystem zur sicheren Speicherung von Credentials	Ja					n/a
Hardwarespeicher zur Ablage von Masterkeys	Ja	Abhängig von Gerät	Ja	Ja	Ja	Ja
Zugriffschutz:						
PIN, Passcodes	Ja	Ja	Ja	Ja	Ja	Ja
Biometrischer Zugriffsschutz	Ja (Touch ID)	Ja (Abhängig von Gerät)	Ja (Abhängig vom Gerät)	Nein	Ja (Iris-basierte Methode bei aktuellen Geräten)	Nein
Applikationsquellen:						
Applikationsstore	Ja	Ja	Ja	Ja	Ja	Ja
Installation von alternativen Quellen	Nein (Ausnahme: Rootzugriff)	Ja (wenn keine weitere Einschränkung durch MDM vorhanden sind)			Nein (außer Entwickleroptionen)	Ja
Installation von Unternehmens- Applikationen aus alternativen Quellen	Ja	Ja	Ja	Ja	Ja	Ja
Updatesituation:						

[1] *Address Space Layout Randomization (ASLR)* bezeichnet die Randomisierung von Speicherbereichen um Angreiferinnen und Angreifern ein Vorhersagen von Speicheradressen zu erschweren.
[2] *Data Execution* Prevention *(DEP)* ermöglicht eine strikte Trennung zwischen ausführbaren Code und nicht ausführbaren Daten. DEP unterbindet die Ausführung von Quellcode im Datenbereich.

Versorgung mit Updates	Sehr gut	Abhängig von Gerätehersteller (hohe Fragmentierung)	Sehr gut	Sehr gut	Sehr gut (garantierter Updatezeitraum von 18 Monaten)	Sehr gut
Unternehmenseinsatz:						
Verfügbare MDM-Regeln	Große Anzahl an Regeln	Wenige Regeln, herstellerspezifische Erweiterungen	Große Anzahl an Regeln	Geringe Anzahl an Regeln (Standard-Android)	Geringe Anzahl an Regeln	Große Anzahl an Regeln für Work Space

2. Einleitung

Seit der Einführung des Apple iPhones im Jahr 2007 entwickeln sich Smartphones sowohl im Privatbereich als auch im professionellen Umfeld zu einem unverzichtbaren Teil unserer modernen Gesellschaft. Durch die Weiterentwicklungen bestehender und Einführung neuer Plattformen sind sowohl Betreiberinnen und Betreiber von IT-Systemen als auch Benutzerinnen und Benutzer gefordert, Überblick über die ständig wachsenden Funktionen zu behalten. Mittlerweile verfügen Smartphones über beinahe dieselben Möglichkeiten zur Datenverarbeitung wie herkömmliche Desktop-Computer und führen so zunehmend auch zur Verarbeitung kritischer Unternehmensdaten am mobilen Gerät. Ein Diebstahl dieser Daten bedeutet somit nicht nur eine Verletzung der Privatsphäre der Benutzerin bzw. des Benutzers, sondern kann bei Offenlegung von Unternehmensdaten einen hohen finanziellen Schaden bedeuten. Diese Entwicklung verdeutlicht die Relevanz umfassender Konzepte zur Sicherstellung des Schutzes von am Smartphone gespeicherten und verarbeiteten Daten.

Dieses Überblickspapier soll der technisch versierten Leserin bzw. dem technisch versierten Leser als Einstieg in die auf den unterschiedlichen Plattformen verfügbaren Sicherheitsmechanismen dienen. Konkret werden neben Verschlüsselung, Zugriffsschutz und der Anbindung an Cloud-Dienste auch die Versorgung mit Updates und die Verteilung von Applikationen betrachtet. Des Weiteren werden die Bring-Your-Own-Device (BYOD) und Mobile-Device-Management (MDM) Möglichkeiten der einzelnen Plattformen vorgestellt, um eine Grundlage für die Bewertung der Eignung einer Smartphone Plattform für den Einsatz im Unternehmen zu bieten. Um den Großteil der eingesetzten Plattformen abzudecken, analysiert dieses Papier die Smartphone-Plattformen Android, iOS, Windows Mobile und BlackBerry, sowie spezifischere Ausprägungen der einzelnen Plattformen. Bei den im Jahr 2015 verkauften Smartphones liefen weltweit bereits 82,8% unter Android, weitere 13,9% unter iOS, gefolgt von Windows Phone mit 2.6% und BlackBerry mit 0,3%[3].

Android ist eine von der Open Handset Alliance unter Führung von Google entwickelte Smartphone-Plattform. Das erste Smartphone unter Android wurde 2008 ausgeliefert, aktuell liegt Version 6 (Marshmallow) der Android-Plattform vor. Apple lieferte 2007 mit dem iPhone eine neue Art von Smartphone aus und stellte die Weichen für zukünftige Entwicklungen im Smartphone-Segment. 2012 setzte Microsoft mit Windows Phone 8 einen Schritt in Richtung einer Vereinheitlichung von Smartphone- und Desktop-Betriebssystemen und veröffentlichte die für mobile Geräte adaptierte Version des Desktop-Betriebssystems Windows 8. Ende 2015 wurde dann die aktuelle Windows 10-Version freigegeben. Um den Entwicklungen der letzten Jahre Folge zu leisten und insbesondere im Unternehmensbereich Marktanteile zurückzuerobern brachte BlackBerry 2013 mit BlackBerry 10 ein überarbeitetes System mit integrierter BYOD-Lösung auf den Markt. In 2015 wurde das von BlackBerry entwickelte Betriebssystem ergänzt durch eine spezielle Android-Version die die Basisfunktionen der BlackBerry-Infrastruktur integriert. Obwohl die einzelnen Smartphone-Plattformen teils unterschiedliche Sicherheitsfunktionen anbieten, können die angebotenen Sicherheitsmechanismen anhand ihrer Kernfunktion gegliedert und verglichen werden. Abschnitt 3 bietet nun eine Einführung zu aktuell auf mobilen Plattformen verfügbaren Sicherheitsmechanismen. Zusätzlich zu den Möglichkeiten der Sicherstellung des Schutzes von Daten werden auch die für einen Unternehmenseinsatz notwendigen Punkte MDM und BYOD betrachtet. Anschließend wird in Abschnitt 4 die Umsetzung der einzelnen Sicherheitsmechanismen auf den Plattformen iOS, Android, Windows Phone 10 und BlackBerry 10 diskutiert. Bei Android wird auf weitere spezifische Ausprägungen mit hoher Relevanz eingegangen: Android for Work, Samsung Knox bzw. BlackBerry PRIV.

[3] IDC, Smartphone OS Market Share 2015 Q2 http://www.idc.com/prodserv/smartphone-os-market-share.jsp, letzter Zugriff am 29.02.2016

3. Sicherheitsmechanismen mobiler Plattformen

Dieser Abschnitt beschreibt die auf Smartphone-Plattformen vorhandenen Sicherheitsfunktionen. Der Leserin bzw. dem Leser soll ein Grundverständnis der jeweiligen Funktion gegeben werden, um eine Grundlage für die in Abschnitt 4 folgende Analyse der Plattformen iOS, Android, Windows Phone 10 und BlackBerry 10 zu bieten.

Konkret werden dabei folgende Punkte diskutiert:

- Basissicherheit
- Verschlüsselung
- Zugriffsschutz
- Applikationsquellen
- Updatesituation
- Cloud-Anbindung
- Mobile-Device-Management (MDM)
- Bring-Your-Own-Device (BYOD)

3.1. Basissicherheit

Viele Sicherheitslücken im Desktopbereich als auch im mobilen Bereich werden durch das Ausnützen sogenannter *Buffer Overflows* ermöglicht. Die Gefahr von *Buffer Overflows* liegt darin, dass ein Angreifer die Adressbereiche in denen bestimmte Daten oder Quellcode im Speicher liegen, vorhersagen kann. Durch Ausnützen von *Buffer Overflows* können so normalerweise nicht zugängliche Daten gelesen, überschrieben oder eingeschleuster Schadcode ausgeführt werden. Als Gegenmaßnahme gilt die *Address Space Layout Randomization (ASLR)*, die Programmen beim Booten oder vor jeder Ausführung Adressbereiche neu zuweist und so ein Vorhersagen von Speicheradressen erschwert. Um das Einschleusen von Quellcode durch einen Angreifer zusätzlich zu erschweren, ermöglicht *Data Execution Prevention (DEP)* eine Unterteilung des Speichers in Daten und Quellcode und unterbindet die Ausführung von Quellcode im Datenbereich.

Als weitere Sicherheitsfunktion unterstützen mittlerweile viele Plattformen einen *Secure-Boot-Prozess*, wobei alle vom System geladenen Module (z.B. Bootloader, Kernel, etwaige Firmware) kryptographisch signiert werden. Die Signatur der einzelnen Module wird während des Boot-Vorgangs überprüft. *Secure Boot* soll die Ausführung eines kompromittierten Betriebssystems unterbinden, da anhand der Signatur Veränderungen im jeweiligen Programmmodul festgestellt werden können.

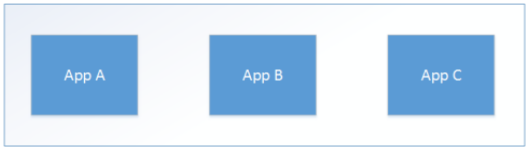

Abbildung 1: Sandboxing-Ansatz

Um zu verhindern, dass Schadsoftware in Form von Applikationen andere Applikationen oder das gesamte Gerät kompromittieren bzw. Daten anderer Applikationen auslesen kann, bieten alle gängigen Smartphone-Plattformen einen Sandboxing-Mechanismus. Abbildung 1 illustriert das Sandboxing-Konzept. Dabei werden Applikationen standardmäßig kaum Rechte eingeräumt, sondern über *Permissions* bzw. Rechte wird von der Benutzerin bzw. vom Benutzer entschieden, ob einer Applikation die angeforderte Funktionalität zur Verfügung gestellt wird. Bei der Implementierung von derartigen Rechtesystemen kann man zwischen Systemen, die bei der Installation um eine Bestätigung bitten, und solchen, welche die Überprüfung zur Laufzeit vornehmen, unterschieden werden. Abschnitt 4 geht auf die Implementierung der genannten Basissicherheitsfunktionen der jeweiligen Plattform näher ein und betrachtet deren Probleme und Limitierungen.

3.2. Zugriffsschutz

Um Geräte vor unbefugtem Zugriff zu schützen, bieten mobile Plattformen unterschiedliche Methoden zum Zugriffsschutz. Diese können von PIN, über alphanumerische Passcodes[4], bis hin zu biometrischen Methoden wie Scan des Fingerabdrucks oder Gesichtserkennung reichen. Bei der Verwendung von Passcodes und PINs ist besonders auf die Länge der Zeichenfolge zu achten, da diese maßgebend für die Dauer der Durchführung einer sogenannten *Brute-Force-Attacke* ist. Unter einer *Brute-Force-Attacke* versteht man das Ausprobieren aller möglichen Eingabewerte, um so einen Passcode zu knacken. Von der Benutzerin bzw. vom Benutzer gewählte Passcodes können auch in die Schlüsselableitung des Verschlüsselungssystems miteinfließen und müssen somit sehr oft im Zusammenhang mit dem Verschlüsselungssystem der Plattform betrachtet werden.

3.3. Verschlüsselung

Die in Abschnitt 4 angeführten Smartphone-Plattformen verfügen über einen Mechanismus zur Verschlüsselung des Dateisystems. Dabei werden alle am Gerät abgelegten Daten verschlüsselt und nach Lösen der Bildschirmsperre wieder entschlüsselt. Unterschiede ergeben sich bei der Speicherung der zur Verschlüsselung verwendeten geheimen Schlüssel. Während mehrere Anbieter über Hardware-Unterstützung verfügen, verwenden andere Lösungen nur vom Passcode der Benutzerin bzw. des Benutzers abgeleitete Schlüssel. Die Tatsache, ob ein Hardware-Element für die Ablage der Schlüssel verwendet wird, hat erheblichen Einfluss auf die Durchführbarkeit von Attacken. Werden Verschlüsselungsschlüssel nicht in der sicheren Umgebung eines Hardware-Elements abgelegt, so besteht die Möglichkeit, Angriffe offline also nicht direkt am Gerät durchzuführen. Dabei können über Rechen-Cluster parallelisiert *Brute-Force-Attacken* durchgeführt werden und so abhängig von der Länge und Komplexität des Passcodes der verwendete Schlüssel berechnet werden. Um solche Attacken zu erschweren bzw. die dafür notwendige Zeit zu erhöhen, werden standardisierte Ableitungsfunktionen für die Berechnung des Schlüssels aus dem Passcode verwendet. Beachtenswert ist die bei Ableitungsfunktionen verwendete Anzahl an Iterationen, da diese maßgeblich für die zeitliche Dauer einer Ableitung ist. Wird die Anzahl der Iterationen zu niedrig gewählt, so findet eine Berechnung im Zuge der Entschlüsselung der Daten zwar innerhalb kurzer Zeit statt und bietet somit eine Erhöhung der Benutzerfreundlichkeit, jedoch vermindert dies auch den zeitlichen Aufwand zur Durchführung einer *Brute-Force-Attacke*. Von einem Hardware-Element abhängige Verschlüsselungssysteme hingegen haben den Nachteil, dass bei einem *Jailbreak*[5] auf die am Gerät vorhandenen Daten in unverschlüsselter Form zugegriffen werden kann, da kein von der Benutzerin bzw. Benutzer gewählter externer Eingabewert in die Verschlüsselung miteingeht.

Verschlüsselungssysteme haben des Weiteren Einfluss auf die verfügbaren Möglichkeiten, Daten am Gerät über eine Remote-Verbindung zu löschen. Durch simples Löschen des zur Verschlüsselung verwendeten kryptographischen Schlüssels, können die Daten nicht mehr entschlüsselt werden und sind somit für Angreifer von keinem Nutzen. Werden Geräte hingegen nicht verschlüsselt, erfordert das Löschen aller am Gerät verfügbaren Daten viel Zeit und wirkt sich erheblich auf den Akkuverbrauch des Geräts aus.

Zusätzlich zur Dateisystemverschlüsselung verfügen einige Plattformen über ein applikationsspezifisches Verschlüsselungssystem. Hierbei können ausgewählte Applikationsdaten zusätzlich verschlüsselt abgelegt werden. Eine Anwendung dieser Sicherheitsfunktionen erfordert jedoch Know-How von der Applikationsentwicklerin bzw. vom Applikationsentwickler. Abschnitt 4 stellt die von den jeweiligen Plattformen verwendeten Verschlüsselungsmethoden vor und diskutiert Möglichkeiten zur zusätzlichen Verschlüsselung von Applikationsdaten.

[4] Anstelle von PIN oder Passwort wird in diesem Dokument die Bezeichnung Passcode verwendet. Dieser umfasst sowohl numerische PINs als auch alphanumerische Passwörter.

[5] Durch Ausnützen einer Sicherheitslücke im Betriebssystem wird uneingeschränkter Zugriff auf das Betriebssystem und dessen Ressourcen erlangt. Jailbreaks ermöglichen das Umgehen von Sicherheitsfunktionen wie z.B. der Bildschirmsperre und können somit eine Entschlüsselung der am Gerät verschlüsselt abgelegten Daten bewirken, falls zur Verschlüsselung kein von der Benutzerin bzw. dem Benutzer gewählter Passcode verwendet wird.

3.4. Applikationsquellen

Alle in Abschnitt 4 betrachteten Plattformen verfügen über einen Applikationsstore, über den Anwendungen am Gerät installiert werden können. Die einzelnen Smartphone-Plattformen unterscheiden sich jedoch stark in der Restriktivität dieser Applikationsstores. Während bei manchen Plattformen Applikationen vor ihrer Veröffentlichung auf verschiedene Aspekte wie Bedienung, Performance und Funktionsweise überprüft werden und die Identität der Entwicklerin bzw. des Entwicklers oder der Anbieterin bzw. des Anbieters über weitere Kanäle (Dokumente etc.) bestätigt sein muss, sehen andere Plattformen keine besondere Prüfung der Identität der Anbieterin bzw. des Anbieters vor. Des Weiteren besteht bei manchen Plattformen die Möglichkeit, die Installation von Applikationspaketen von Drittanbietern zuzulassen und somit auch alternative Applikationsquellen zu ermöglichen. Eine gängige Vorgehensweise ist es, Applikationspakete kryptographisch zu signieren und eine Installation von nur vom Hersteller der Plattform signierten Applikationen aus dem offiziellen Applikationsstore zu erlauben. Vorteil einer restriktiveren Applikationsverteilung ist das Unterbinden von *Sideloading*, wobei Geräte über Downloads oder über via E-Mail verteilte Applikationspakete mit Malware infiziert werden. Um im Nachhinein als bösartig erkannte Applikationen wieder von mobilen Geräten zu entfernen, bieten die gängigen Plattformen des Weiteren eine sogenannte *Remote Kill* Funktionalität an.

3.5. Updates

Die Verfügbarkeit von Updates übt entscheidenden Einfluss auf die Sicherheit der mobilen Plattform aus, da im Zuge von Updates Sicherheitslücken geschlossen und unter Umständen neue Sicherheitsfeatures ausgeliefert werden. Zwischen den unterschiedlichen Smartphone-Plattformen ergeben sich nicht nur Unterschiede in den Updatezeiträumen, sondern auch bei der Auslieferung der Updates. Diese können direkt vom Anbieter der Plattform unverändert weitergegeben werden. Alternativ erhalten bei manchen Anbietern Mobilfunknetzprovider die Möglichkeit, Updates zu verzögern, zu modifizieren oder gänzlich nicht an die Kundin oder den Kunden weiterzugeben.

3.6. Cloud-Anbindung

Cloud-Dienste erlauben es Benutzerinnen und Benutzern, Daten zentral abzulegen und oftmals auch vollständige Backups ihrer Geräte anzulegen. Beachtenswert ist hierbei die Sicherheit der abgelegten Daten und ob diese durch geeignete kryptographische Methoden vor nicht legitimem Zugriff geschützt sind. Da Cloud-Dienste auch von Drittanbietern zur Verfügung gestellt werden, wird in diesem Papier nur auf die generellen Möglichkeiten und die vom Hersteller der Smartphone-Plattform angebotene Standardlösung eingegangen.

3.7. MDM

Mobile-Device-Management (MDM) bezeichnet das zentrale Steuern von Funktionalitäten und Konfigurationen am Gerät. Über direkt im Betriebssystem integrierte oder von Drittanbietern entwickelte Schnittstellen können IT-Administratorinnen und IT-Administratoren Sicherheitsfunktionen am Gerät aktivieren und u.a. bei Verlust eines Geräts Inhalte per Remote-Verbindung löschen (*Remote Wipe*). MDM-Lösungen finden hauptsächlich in Unternehmen Verwendung, um die an Mitarbeiterinnen und Mitarbeiter ausgegebenen Geräte zu steuern. Die angebotene Funktionalität und die Integration im mobilen Gerät unterscheiden sich stark zwischen den einzelnen Plattformen. Ein wichtiger Punkt bei der Betrachtung von MDM-Lösungen ist, ob Benutzerinnen und Benutzer die Möglichkeit besitzen, durch eigenständiges Entfernen oder Deaktivieren der MDM-Software das Gerät der zentralen Kontrolle zu entziehen und weiterhin auf vom Unternehmen bereitgestellte Daten zuzugreifen.

3.8. BYOD

Als *Bring-Your-Own-Device* (BYOD) wird das Einbringen eines privaten mobilen Endgeräts in ein Unternehmen bezeichnet. Dabei befindet sich das Gerät im Besitz der Mitarbeiterin oder des Mitarbeiters und unterliegt somit nicht gänzlich der Kontrolle des Unternehmens. Zusätzlich zu

rechtlichen und organisatorischen Herausforderungen, birgt das BYOD-Szenario auch technische Probleme. Durch die Tatsache, dass Mitarbeiterinnen und Mitarbeiter Smartphones unterschiedlicher Hersteller und mit unterschiedlichen Betriebssystemen in das Unternehmen einbringen möchten, muss die IT-Administration über die nötige Infrastruktur und das für die unterschiedlichen Smartphone-Plattformen nötige technische Knowhow verfügen. Des Weiteren verfügen nicht alle Plattformen über im Betriebssystem integrierte BYOD-Features, die eine saubere Trennung von dienstlichen und privaten Daten zulassen. Um zusätzlich im Unternehmen entwickelte und für Unternehmensprozesse erworbene Applikationen an die Geräte der Mitarbeiterinnen und Mitarbeiter zu verteilen bieten mehrere Hersteller *Mobile-Application-Management* Lösungen. Abbildung 2 bietet einen Überblick über die unterschiedlichen BYOD-Szenarien.

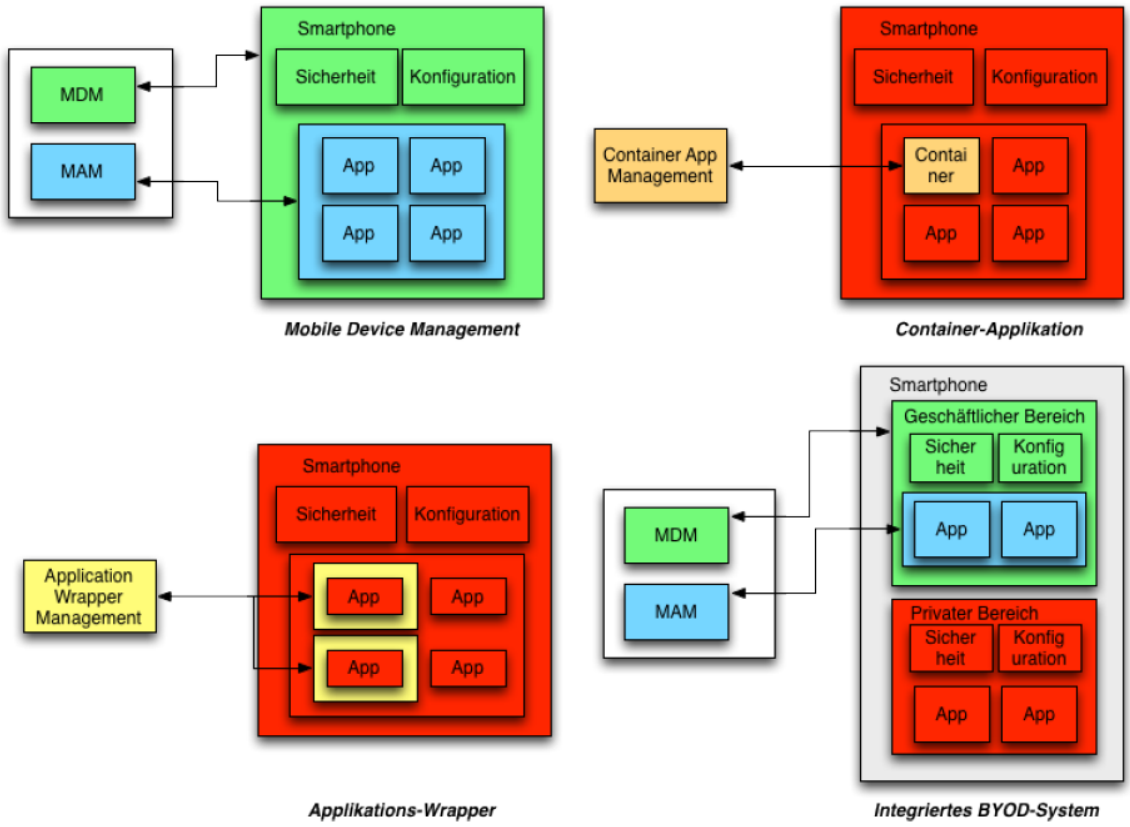

Abbildung 2: BYOD-Szenarien

Um dem Problem unterschiedlicher, im Unternehmen eingesetzter Betriebssysteme entgegen zu wirken, wurden wie in Abbildung 2 dargestellt innerhalb der letzten Jahre von Drittanbietern entwickelte *Container-Applikationen* und *Applikations-Wrapper* ausgeliefert. Eine Container-Applikation beinhaltet üblicherweise einen E-Mail-Dienst, Kalender, etc. und implementiert plattformunabhängige Sicherheitsfunktionen. In aktuellen Betriebssystemen werden typischerweise Basisfunktionen bereit gestellt, die es ermöglichen Container zur Verfügung zu stellen, die einen erhöhten Schutz bieten, da die zugrundeliegenden Sicherheitsfunktionen direkt im Betriebssystem integriert sind. Applikations-Wrapper hingegen erweitern installierte Applikationen um Sicherheitsfunktionen. Da zur Modifizierung von Applikationen jedoch das Applikationspaket zugänglich sein muss, sind Applikationswrapper nur beschränkt in der Praxis einsetzbar. Die restriktivste Form BYOD zu betreiben, ist das in Abbildung 2 oben links abgebildete Steuern des Geräts über eine MDM-Lösung. Abhängig von der Plattform können über die verwendete MDM-Software Applikationen an die Geräte verteilt werden bzw. kann die Erstellung von Whitelists für bestimmte Applikationen und Applikationsquellen vorgenommen werden. Da gängige MDM-Lösungen jedoch stark in die Privatsphäre der Benutzerinnen und Benutzer eingreifen, den Funktionsumfang des Geräts einschränken und unter Umständen auch zu einem Verlust der privaten am Smartphone vorhandenen Daten führen können, ist diese Form hauptsächlich abhängig von der Akzeptanz durch die Benutzerin bzw. den Benutzer.

4. Plattformanalyse

In diesem Abschnitt werden die Sicherheitsfunktionen der Plattformen iOS, Windows 10 Mobile, BlackBerry 10, Android, Android – BlackBerry PRIV und Android - Samsung KNOX analysiert.

4.1. iOS

Analysierte Version	9.3.x
Letzte Aktualisierung	Mai 2016

Apple brachte 2007 das iPhone und das dazugehörige Betriebssystem iOS auf den Markt und hat damit eine neue Ära im Bereich der Mobiltechnologie eingeleitet. Mittlerweile wird iOS auch auf anderen mobilen Geräten wie dem iPad oder dem iPod Touch eingesetzt. Apple veröffentlicht jährliche Updates von iOS, zuletzt wurden im September 2015 iOS 9 und damit auch die neuen Flagship-Modelle iPhone 6s und iPhone 6s Plus auf den Markt gebracht. Die derzeit aktuellste iOS Version ist iOS 9.3.2 und wurde am 16 Mai 2016 veröffentlicht. Das iPhone zeichnet sich im Vergleich zu Android mit einer restriktiveren Applikationspolitik aus, wohingegen Android offen für Modifikationen von Geräteherstellern ist.

Im nachfolgenden Abschnitt werden Sicherheitsfunktionen von iOS näher betrachtet.

4.1.1. Basissicherheit

Um die Integrität des Betriebssystems zu gewährleisten, kommt bei iOS ein *Secure Boot* Mechanismus zur Anwendung. Dabei wird zur Boot-Zeit des Systems schrittweise die kryptographische Signatur aller zu ladenden Softwarekomponenten überprüft. Sollte in einem Schritt des Boot-Prozesses die Verifizierung der Signatur einer Softwarekomponente fehlschlagen, wird der Systemstart abgebrochen und das Gerät in den *Recovery Mode* gebootet. Nur durch Zurücksetzen des Systems in den Werkszustand ist eine erneute Verwendung des Geräts möglich. Über Secure Boot wird gewährleistet, dass nur von Apple gelieferte und nicht modifizierte Komponenten vom System geladen werden [1]. Bei Geräten mit Mobilfunkunterstützung wird zusätzlich ein ähnlicher Secure-Boot-Prozess durch den Baseband-Prozessor ausgeführt. Hierbei wird die Integrität und Echtheit der Baseband-Software verifiziert. Bei Geräten mit einem A7-Prozessor oder einem Nachfolgermodell führt der *Secure Enclave* Coprozessor zusätzlich einen Secure-Boot-Prozess aus, um dessen Software zu verifizieren. Die Secure-Boot Prozesse stellen sicher, dass nur vertrauenswürdige unveränderte Software installiert werden kann. Um zu verhindern, dass ein Angreifer eine ältere Softwareversion von iOS oder von der Software für den *Secure Enclave* Coprozessor installiert, die eine möglicherweise in neueren Versionen bereits geschlossene Sicherheitslücke enthält, verwendet iOS einen Prozess namens *System Software Authorization*. Bei einem Update wird ein Apple Server kontaktiert, der den Updatevorgang autorisieren muss. An diesen Server werden neben Informationen über alle zu installierenden Softwarepakete eine Zufallszahl und die eindeutige ID des Gerätes geschickt. Der Server kontrolliert die Information der zu installierenden Software gegen eine Liste von erlaubter Software und schickt, falls die Installation genehmigt wird eine signierte Antwort. Durch die Zufallszahl wird verhindert, dass ältere aufgezeichnete Serverantworten verwenden werden können. Durch die eindeutige ID des Gerätes wird verhindert, dass Serverantworten für andere Geräte missbräuchlich verwendet werden können.

Seit Apple iOS 5 werden die Adressbereiche von systeminternen Applikationen und Systembibliotheken zufällig zugewiesen. Applikationen von Drittanbietern verwenden standardmäßig ASLR, jedoch steht es der Entwicklerin bzw. dem Entwickler frei, ASLR zu deaktivieren. Die über ASLR gebotene zufällige Zuweisung der Adressbereiche erschwert es Angreifern, Speicheradressen vorherzusagen, um z.B. eingeschleusten Quellcode auszuführen. Zusätzlich wird über den im ARM-Prozessor verfügbaren *Execute Never (XN)* Mechanismus (die sogenannte „Data Execution Prevention“) eine Unterteilung in ausführbaren Quellcode und nicht ausführbare Daten erreicht. Eine Ausnahme bildet der Speicherbereich der mobilen Version von Safari, um eine *Just-in-Time-Kompilierung* von Javascript im Browser zu ermöglichen [1].

ASLR und DEP schützen somit das System vor eingeschleustem Schadcode. Des Weiteren wird über den Sandboxing-Mechanismus eine Isolierung von Applikationen erreicht. Sowohl Applikationen von Drittanbietern als auch vorinstallierte Applikationen können lediglich auf Daten im eigenen Speicherbereich zugreifen. Ein direkter Zugriff auf das Dateisystem des Geräts ist nicht möglich. Sandboxing unterbindet somit den Zugriff von böswilligen Applikationen (Schadsoftware) auf Daten anderer Applikationen und damit eine Kompromittierung des gesamten Systems.

4.1.2. Verschlüsselung

Apple iOS unterscheidet zwischen zwei Verschlüsselungssystemen, der standardmäßig aktiven Verschlüsselung des Dateisystems und einem applikationsspezifischen Verschlüsselungssystem zum Schutz einzelner Dateien und von *Keychain-Einträgen*. Die *Keychain* bezeichnet einen Mechanismus zur geschützten Ablage von Zugangsdaten wie Benutzernamen, Passwörtern, Zertifikaten, etc. am Gerät. Abbildung 3 bietet eine schematische Darstellung der beiden Verschlüsselungssysteme. Nachfolgend werden die beiden Systeme näher betrachtet.

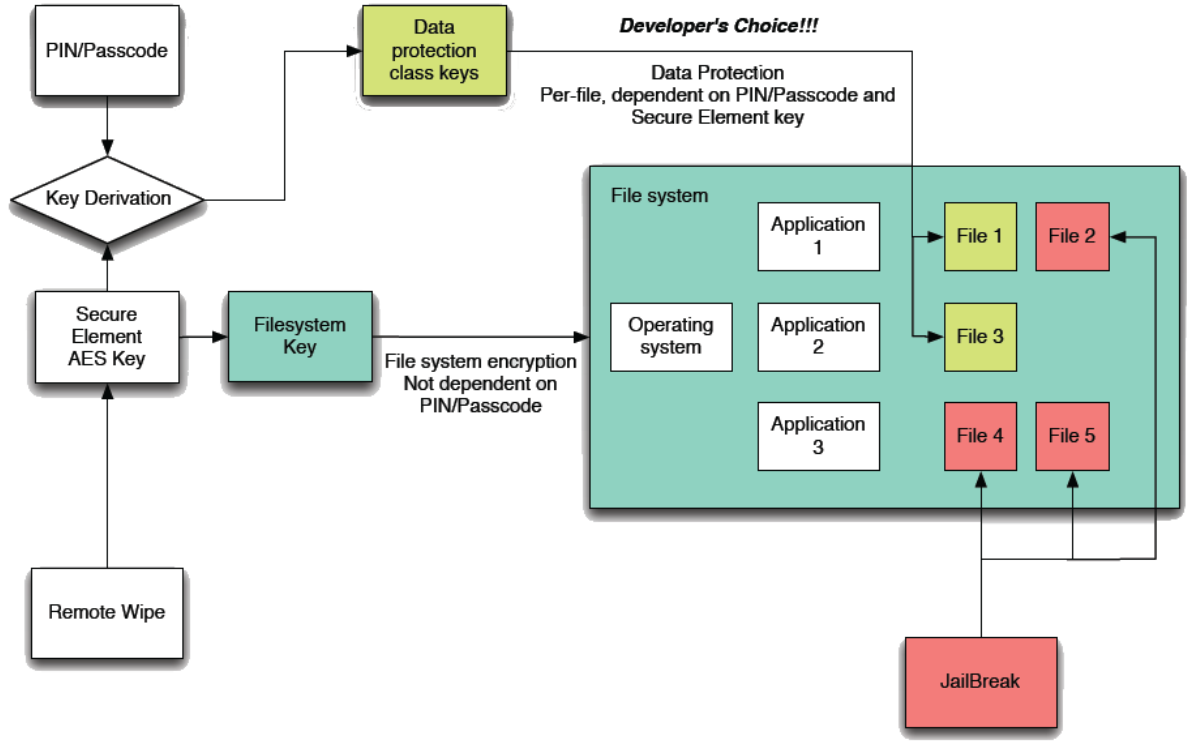

Abbildung 3: Übersicht iOS Verschlüsselung – Vereinfachte Darstellung

Dateisystemverschlüsselung

Apple-Geräte beinhalten seit der Einführung des iPhone 3GS ein Hardware-Element, welches einen für jedes Gerät eindeutigen kryptographischen Schlüssel enthält. In neueren Versionen entspricht dieses Hardware-Element der „*Secure Enclave*“ die sicherheitskritische Operationen ausführt. Die Datenpartition eines iOS-Gerätes wird mit einem aus diesem Schlüssel abgeleiteten Schlüssel verschlüsselt. Im Detail verfügt jedes Hardware-Element über einen eindeutigen, 256 Bit langen AES Schlüssel, dem *UID Key*. Dieser schützt den im flüchtigen Speicher abgelegten *Dateisystem Master Schlüssel* (*EMF Key*). Der *EMF Key* wird direkt zur Verschlüsselung aller Daten am Dateisystem genutzt. Im Falle eines Diebstahls des Geräts kann die für eine Remote-Löschung benötigte Zeit durch ein Entfernen des *EMF Keys* anstelle aller am Gerät vorhandenen Daten signifikant vermindert werden.

Die Verwendung des Hardware-Elements zur Schlüsselableitung bietet zum einen den Vorteil, dass ein Entschlüsseln der Datenpartition nur am Gerät ermöglicht wird. Zu beachten ist jedoch, dass

diese standardmäßige aktiviere Verschlüsselung nicht vom PIN-Code des Benutzers abhängt. Um hier einen zusätzlichen Schutz für einzelnen Dateien und KeyChain-Einträge (Passwörter, Schlüssel) zu bieten steht ein zweites Verschlüsselungssystem zur Verfügung, das vom Applikationsentwickler je nach Anwendung verwendet werden kann.

Verschlüsselung einzelner Dateien bzw. Keychain Einträge

Zusätzlich zur Dateisystemverschlüsselung können von der Entwicklerin bzw. vom Entwickler seit iOS 4 durch Angabe von sogenannten *Protection Classes* ausgewählte Dateien zusätzlich verschlüsselt werden. Eine *Protection Class* definiert den Zeitpunkt der Ver- und Entschlüsselung der ausgewählten Datei. Die für die *Protection Classes* verwendeten Schlüssel werden vom in der sicheren Umgebung des Hardware-Elements enthaltenen Schlüssel und vom von der Benutzerin bzw. vom Benutzer gewählten Passcode abgeleitet. Für jede *Protection Class* existiert ein eigener *Class Key*, der die für jede zu schützende Datei eindeutigen Schlüssel schützt. In iOS 9 stehen die folgenden *Protection Classes* zur Verfügung, die vom Entwickler je nach Anwendung korrekt für das Speichern von Daten verwendet werden müssen:

Data Protection Classes – für die Verschlüsselung von Dateien:

- **NSProtectionComplete**: Dateien dieser Klasse sind nur im entsperrten Zustand zugänglich d.h. die Schlüsselableitung erfolgt nach dem Entsperren des Geräts und die abgeleiteten Schlüssel werden beim Sperren des Geräts aus dem Speicher entfernt. Dateien dieser Klasse sind somit Hintergrundtasks nicht zugänglich.
- **NSFileProtectionCompleteUntilFirstUserAuthentication**: Diese Klasse verhält sich grundsätzlich ident zu *NSProtectionComplete*, mit dem Unterschied, dass der *Protection Class Key* nach Einsetzen der Bildschirmsperre weiterhin verfügbar ist. Die Entschlüsselung der Daten erfolgt nach dem ersten Entsperren des Geräts nach einem Neustart.
- **NSFileProtectionCompleteUnlessOpen**: Diese Klasse wird typischerweise für Dateien benutzt, die im gesperrten Zustand geschrieben werden, z.B. Verschlüsselung einer eingehenden E-Mail Nachricht. Dabei wird ein asymmetrisches Schlüsselpaar generiert. Der private Schlüssel wird vom symmetrischen *Protection Class Key* geschützt und mit Hilfe des öffentlichen Schlüssels können somit eingehende Daten verschlüsselt werden.
- **NSProtectionNone**: Dateien dieser Klasse nutzen lediglich die Dateisystemverschlüsselung. Diese Klasse umfasst alle Dateien, für die keine der oben genannten Klassen gesetzt wurde.

iOS ermöglicht die Speicherung von privaten Schlüsseln, Passwörtern und Zertifikaten in der sogenannten *Keychain*. Dabei handelt es sich um einen Speicher der es Applikationen ermöglicht, Zugangsdaten sicher abzulegen. Zugang zu *Keychain-Einträgen* wird nur Applikationen derselben Entwicklerin bzw. desselben Entwicklers gewährt, d.h. Applikationen die mit demselben kryptographischen Schlüssel der Entwicklerin bzw. des Entwicklers signiert sind. Für *Keychain-Einträge* existieren den *Protection Classes* ähnliche Kategorien. Die Ableitung der Schlüssel erfolgt ident zur Verschlüsselung einzelner Dateien.

KeyChain Data Protection Classes – für das sichere Speichern von *KeyChain*-Einträgen:

- **kSecAttrAccessibleWhenUnlocked**: Gleiches Verhalten wie *NSProtectionComplete.*
- **kSecAttrAccessibleAfterFirstUnlock**: Gleiches Verhalten wie *NSFileProtectionCompleteUntilFirstUserAuthentication***.**
- **kSecAttrAccessibleAlways**: Gleiches Verhalten wie *NSProtectionNone*.
- **kSecAttrAccessible-WhenPasscodeSetThisDeviceOnly**: Einträge mit dieser Klasse werden nur in der *KeyChain* abgelegt, wenn ein Passcode für die Sperrung des Geräts definiert wurde. Ansonsten entspricht das Verhalten der Klasse *kSecAttrAccessibleWhenUnlocked.*

Da ein Hardware-Element in die Schlüsselableitung miteinfließt, können die Daten und die *Keychain-Einträge* nur am Gerät selbst entschlüsselt werden. Die zusätzliche Abhängigkeit vom Passcode bietet auch im Falle eines *Jailbreaks* Schutz vor nicht legitimen Zugriffen. Um dennoch Zugriff zu den verschlüsselten Daten zu erhalten, ist die Durchführung einer *Brute-Force-Attacke* am Gerät erforderlich. Seit der Integration des *Secure Enclave* Systems sind Brute-Force Angriffe noch schwerer durchzuführen. Bei der Ableitung des Schlüssels aus dem Passcode wird der *Secure Enclave* Prozessor benötigt. Kommt es zu einer falschen Passcode-Eingabe und somit zum falschen Ableitungsergebnis des Schlüssels, werden Verzögerungen vom *Secure Enclave* Prozessor eingeführt, die die Anzahl der Versuche pro Zeiteinheit reduzieren. Die Dauer der Verzögerung wird mit jedem Fehlversuch erhöht. Vor der Einführung des *Secure Enclave* Komponenten wurden diese Verzögerungen vom Betriebssystem (in Software) realisiert und konnten damit leichter umgangen werden.

Es muss angemerkt werden, dass der Schutz durch die *Protection Classes* explizit von der Entwicklerin bzw. vom Entwickler aktiviert werden muss. Seit iOS 7 wird *NSFileProtectionCompleteUntilFirstUserAuthentication* als Standardklasse für Daten von Drittanbieterapplikationen verwendet, sofern den Daten keine andere Klasse zugewiesen wird. Die Benutzerin bzw. der Benutzer erhält keine Information, ob und wie eine Applikation die Daten zusätzlich verschlüsselt. Des Weiteren können Daten zwar von einer Applikation z.B. einem E-Mail Client verschlüsselt und somit geschützt werden, jedoch kann nicht sichergestellt werden, dass beispielsweise die am Gerät vorhandene Applikation zur Anzeige von PDF-Dateien, die aus Anhängen geöffneten Dateien korrekt verschlüsselt und somit auch im Falle eines *Jailbreaks* schützt.

4.1.3. Zugriffsschutz

Benutzerinnen oder Benutzer können beliebige 4-stellige oder 6-stellige Ziffernfolgen oder beliebig lange alphanumerische Passcodes setzen, um das Gerät vor unerlaubten Zugriffen zu schützen. Der von der Benutzerin bzw. vom Benutzer gewählte Passcode fließt zusätzlich in die Ableitung des Schlüssels für das applikationsspezifische Verschlüsselungssystem ein. Die Sicherheit dieses Verschlüsselungssystems ist jedoch hauptsächlich abhängig von der Komplexität des gewählten Passcodes. Um auch komplexe und für die häufige Eingabe nicht praktikable Passcodes zu ermöglichen, wurde mit dem iPhone 5s mit *Touch ID* ein biometrisches Verfahren zum Entsperren des Geräts eingeführt. Dabei muss von der Benutzerin bzw. vom Benutzer einmalig nach jedem Reboot der gewählte Passcode angegeben werden. Anschließend kann das Gerät über den im Home-Button integrierten Fingerabdruck-Sensor entsperrt werden. Der Fingerabdruck des Benutzers wird in der sogenannten *Secure Enclave* gespeichert [1]. Diese bezeichnet einen im iPhone 5s und neueren Geräten verbauten Coprozessor mit einem eigenen verschlüsselten Speicherbereich. Bei der Herstellung erhält jede *Secure Enclave* eine eigene *Unique ID*, die auch in die spätere Ableitung des Schlüssels zur Verschlüsselung des Speicherbereichs der *Secure Enclave* miteinfließt. Beim ersten Entsperren des Geräts nach dem Bootvorgang, werden die Data Protection-Schlüssel für die Klasse *Complete* mit dem nur in der *Secure Enclave* verfügbaren Schlüssel verschlüsselt. Bei nachfolgenden Entsperrvorgängen wird zuerst der Fingerabdruck auf Übereinstimmung geprüft und anschließend der Schlüssel für die Data Protection Klasse *Complete* mit dem in der *Secure Enclave* verfügbaren Schlüssel entschlüsselt und dem System zur Verfügung gestellt. Spätestens nach 48 Stunden oder nach fünf erfolglosen Versuchen das Gerät über *Touch ID* zu entsperren, oder wenn eine Remote-Sperrbefehl empfangen wird, werden die verschlüsselten Data Protection Schlüssel gelöscht und zum Entschlüsseln der Daten ist eine erneute Eingabe des Passcodes erforderlich.

4.1.4. Applikationsquellen

Der AppStore stellt prinzipiell die einzige Möglichkeit dar, Applikationen auf iOS Geräten zu installieren[6]. Für Entwicklerinnen und Entwickler besteht jedoch die Möglichkeit, zu Testzwecken Applikationen auf ihren Entwicklungsgeräten zu installieren. Unternehmen können dabei über sogenannte *Provisioning Profile* unternehmensinterne Applikationen an ihre Mitarbeiterinnen und Mitarbeiter verteilen[7]. Dazu ist jedoch eine Registrierung eines eigenen Unternehmenskontos bei Apple notwendig.

4.1.5. Updatesituation

Generell existiert bei iOS Geräten im Vergleich zu Android kaum Fragmentierung. Die im September 2015 veröffentlichte Betriebssystem-Version iOS 9 wird zum momentanen Zeitpunkt bereits von 79% Prozent aller Geräte genutzt[8]. Apple bietet Benutzerinnen und Benutzern eine langfristige Versorgung mit Updates. Apple ermöglicht somit Updates auf iOS 9 auch für bereits 2011 erschienene iPhone 4s Geräte. Für das 2010 veröffentlichte iPhone 4 steht hingegen nur iOS bis Version 7.1.2 zur Verfügung.

4.1.6. Cloud-Anbindung

iOS-Geräte können für unterschiedliche Funktionen die Apple Cloud-Lösung „iCloud“ verwenden. Ein essentieller Punkt bei der Verwendung ist der Schutz des iCloud Kontos, der von den Sicherheitseigenschaften des Benutzerkontos abhängt (Eigenschaften des Passworts, Verwendung einer 2-Faktor Lösung).

Backup:

iOS ermöglicht über iTunes ein vollständiges Backup des Dateisystems. Standardmäßig sind alle Dateien, auch Applikationsdaten im Backup enthalten. Entwicklerinnen bzw. Entwicklern wird jedoch die Möglichkeit geboten, während der Entwicklung der Applikation zu entscheiden, ob Dateien vom Backup exkludiert werden sollen. Dazu werden die gewünschten Dateien von der Entwicklerin bzw. vom Entwickler markiert.

Generell kann man bei iOS zwischen verschlüsselten und unverschlüsselten Backups unterscheiden. Unverschlüsselt bedeutet in diesem Zusammenhang, dass durch die Benutzerin bzw. den Benutzer kein separates Backup-Passwort gesetzt wurde, sondern die Standardfunktionalität des Betriebssystems genutzt wird.

- **Verschlüsseltes Backup.** Wird von der Benutzerin bzw. vom Benutzer die Erstellung eines verschlüsselten iTunes Backup gewünscht, so muss diese/r ein Backup-Passwort festlegen. Aus dem Backup-Passwort wird ein kryptographischer Schlüssel (*Backup Key*) mittels der PBKDF2-Funktion (10000 Iterationen) abgeleitet, der in weiterer Folge die Daten verschlüsselt. Verschlüsselte Backups ermöglichen eine Wiederherstellung der *System-Keybag* (enthält alle *Keychain-Einträge* des Geräts) auf einem anderen Gerät, wohingegen bei unverschlüsselten Backups eine Wiederherstellung dieser nur am selben Gerät erfolgen kann. Verschlüsselte Backups bieten zusätzliches Angriffspotenzial in der Form von *Brute-Force-Attacken*, indem Angreifer versuchen, das von der Benutzerin bzw. vom Benutzer gewählte Backup-Passwort zu erraten, um so Zugriff auf die *Keychain-Einträge* und restlichen Dateien zu erlangen.

- **Unverschlüsseltes Backup.** Bei unverschlüsselten Backups wird die *System-Keybag*, die die *Keychain-Einträge* enthält, mit dem im Hardware-Element vom *UID Key* abgeleiteten Schlüssel verschlüsselt abgelegt. Somit ist eine Entschlüsselung nur am selben Gerät

[6] iOS Geräte mit Jailbreak umgehen diese Restriktion und ermöglichen auch eine Installation von Applikationen aus alternativen Quellen. Der CydiaStore beispielsweise bietet eine Vielzahl an Applikationen für Geräte mit Jailbreak.
[7] iOS Developer Enterprise Program https://developer.apple.com/programs/ios/enterprise/, letzter Zugriff am 8.4.2014
[8] Laut https://developer.apple.com/support/appstore/ laufen bereits 79% aller iOS Geräte unter iOS 9 und weitere 16% unter iOS 8. Daten vom 11. März 2016.

möglich. Dies führt zu der Situation, dass *Keychain-Einträge* bei unverschlüsselten Backups besser gesichert sind als bei Backups mit aktivierter Verschlüsselung. Angreifer erhalten jedoch bei unverschlüsselten Backups durch Zugriff auf das Backup (z.B. durch unerlaubtes Kopieren) Zugang zu den am Gerät befindlichen Dateien. Zu beachten ist, dass auch Dateien, die am Gerät durch eine *Protection Class* geschützt sind, im Klartext im iTunes Backup enthalten sind.

Zusätzlich können Backups auf Apples Cloud-Dienst iCloud abgelegt werden. Es können somit analog zum Backup über iTunes fürs Backup markierte Daten synchronisiert werden. Folgende Daten sind im iCloud Backup enthalten [1]:

- Informationen über gekaufte Filme, Musik, TV-Serien, Apps und Bücher
- Fotos und Videos aus der Camera Roll App
- Kontakte, Kalendereinträge, Erinnerungen und Notizen
- Geräteeinstellungen
- App Daten
- Nicht gekaufte, aber zu iBooks hinzugefügte PDFs und Bücher
- Anrufsliste
- Home Screen und App Anordnung
- iMessages, SMS und MMS
- Klingetöne
- HomeKit und Healthkit Daten
- Visual Voicemail

Die Daten werden in ihrer originalen Form abgelegt, dies bedeutet verschlüsselte Daten werden auch in der iCloud verschlüsselt abgelegt. Die dazugehörenden Protection Class Schlüssel werden mit iCloud Schlüsseln geschützt. Dateien mit der *Protection Class No Protection* werden zwar während der Übertragung verschlüsselt, aber unverschlüsselt gespeichert.

Neben dem iCloud bzw. dem iTunes Backup bietet iOS seit der Version 7.0.3 auch noch eine iCloud KeyChain an. Diese kann zur Synchronisierung von Schlüsseln und Passwörtern zwischen mehreren iOS und Mac-Geräten benutzt werden. Apple hat keinen Zugriff auf die synchronisierten Daten.

Mit iOS5 hat Apple einen virtuellen Assistenten namens Siri eingeführt. Siri steht für **S**peech **I**nterpretation and **R**ecognition **I**nterface. Wie der Name andeutet handelt es sich dabei um einen mittels natürlicher Sprache gesteuerten Assistenten, welcher einfache Aufgaben erledigen kann. Einige Funktionen, wie das Vorlesen von Kurznachrichten, oder die Aktivierung per Sprache werden am Gerät selbst ausgeführt. Bei Anfragen an den Apple Server, wird der Vorname, der Nachnahme sowie je nach Anfrage die ungefähre oder die genaue Position mitgeschickt. Übermittelte Audiodaten werden 6 Monate lang personalisiert gespeichert und danach bis zu 2 Jahre lang anonymisiert. Um möglichst viele Anfragen abdecken zu können, hat Siri unter anderem auf die Musiksammlung (Liedernamen, Playlisten, Artisten, usw.), auf die Kontakte sowie auf die Namen der Erinnerungslisten Zugriff.

4.1.7. MDM

Um den Sicherheitsanforderungen von Unternehmen gerecht zu werden, verfügt Apple iOS über einen bereits im Betriebssystem integrierten MDM-Client[9]. Die Durchsetzung von Sicherheitsrichtlinien erfolgt entweder über das von Apple angebotene *Apple Configurator Tool* oder über MDM-Software von Drittanbietern. Durch die Installation eines vom Unternehmen bereitgestellten Konfigurations-Profils wird die MDM-Funktionalität am Gerät aktiviert. iOS verfügt über eine Vielzahl an Konfigurationsmöglichkeiten um das Gerät an die Sicherheitsbedürfnisse des Unternehmens anzupassen. Folgende Aufzählung bietet einen Überblick über die unter iOS zur Verfügung stehenden MDM-Regeln [2]:

- **Zugriffsschutz**
  - Remote Wipe aller Daten am Gerät
  - Erzwingen der Länge und Komplexität des verwendeten Passcodes
  - Festlegung der Verwendungsdauer eines Passcodes, bzw. nach welchem Zeitraum ein neuer Passcode am Gerät festgelegt werden muss
- **Einschränkungen der Funktionalität**
  - Unterbinden der Verwendung der Kamera am mobilen Gerät
  - Unterbinden der Erstellung von Screenshots
- **Applikationen**
  - Unterbinden der Möglichkeit, Applikationen zu installieren
  - Deaktivierung von Apples sprachgesteuerten persönlichen Assistenten Siri
  - Einschränkungen der Verwendung bestimmter Applikationen, z.B. Sperren von YouTube oder des iTunes Stores
  - Unterbinden der Verwendung von Safari
  - Anpassung der sicherheitsrelevanten Einstellungen von Safari, z.B. Javascript deaktivieren, Unterbinden von Pop-ups oder Cookies etc.
  - Remote-Installation von unternehmensinternen Applikationen
- **Cloud-Anbindung**
  - Unterbinden des Synchronisierens von Dokumenten auf iCloud
  - Unterbinden von Backups auf iCloud
- **Sicherheit**
  - Erzwingung der Ablage verschlüsselter Backups (betrifft iTunes und iCloud)
  - Unterbinden, dass der Benutzer nicht vertrauenswürdige Zertifikate akzeptieren kann
  - Deaktivieren des Sendens von Diagnosedaten (z.B. nach Systemfehlern) an Apple
- **Bereitstellung von Konfigurationsdaten[10]**
  - Konfiguration von VPN oder WLAN
  - E-Mail und Exchange Einstellungen
  - Bereitstellung von Zugangsdaten, Zertifikaten etc.

Über den *Apple Configurator* bzw. das *iPhone Configuration Utility* wird des Weiteren spezifiziert, ob die Benutzerin bzw. der Benutzer das Konfigurationsprofil entfernen kann (*Entfernung ohne Autorisierung*, *Entfernung mit Autorisierung* oder *Entfernung nicht möglich*).

4.1.8. BYOD

Bring-Your-Own-Device bezeichnet den Gebrauch des eigenen mobilen Geräts, um Zugriff auf Unternehmensdaten oder Services zu erhalten. iOS verfügt über keine im Betriebssystem integrierte BYOD-Lösung. MDM bietet eine Möglichkeit BYOD auf iOS-Geräten zu erlauben. Benutzerinnen

[9] Detaillierte Informationen zu MDM als auch BYOD bzw. generell den Einsatz von iOS Geräten im Unternehmensumfeld können der Website von Apple entnommen werden: https://www.apple.com/support/business-education/mdm/, letzter Zugriff am 14.03.2016

[10] https://developer.apple.com/library/ios/featuredarticles/iPhoneConfigurationProfileRef/Introduction/Introduction.html , letzter Zugriff am 11.04.2014

und Benutzer müssen dazu das jeweilige Konfigurationsprofil des Unternehmens installieren. Dieses kann jedoch später nach Belieben wieder eigenständig entfernt werden[11].

[11] https://www.apple.com/iphone/business/it/management.html , letzter Zugriff am 11.04.2014

4.2. Windows 10 Mobile

Analysierte Version	10
Letze Aktualisierung	Mai 2016

Windows 10 Mobile wurde im November 2015 eingeführt, es handelt sich dabei um den Nachfolger von Windows Phone 8.1. Windows 10 Mobile wurde stark an die Desktop Version angelehnt und benutzt auch denselben Kernel.

In diesem Abschnitt werden die von der Windows 10 Mobile Plattform angebotenen Sicherheitsfunktionen näher beschrieben und Möglichkeiten zum Mobile-Device-Management und zur Realisierung von BYOD-Szenarien diskutiert.

4.2.1. Basissicherheit

Windows 10 Mobile bietet so wie Android, iOS und BB10 standardmäßig die Basissicherheitsfunktionen *Address-Space-Layout-Randomization* (ASLR) und *Data-Execution-Prevention (DEP)*. Um zu verhindern, dass Schadsoftware andere am Gerät installierte Applikationen kompromittiert und sicherheitskritische Daten ausliest, laufen Windows 10 Mobile Applikationen und Teile des Betriebssystems in sogenannten *AppContainern. Dabei handelt es sich um eine isolierte Sandbox,* die standardmäßig über minimale Rechte verfügt. Über *Capabilities*[12] bzw. das Rechtesystem von Windows Mobile 10 können diese Sandboxes um Funktionen erweitert werden, z.B. Zugriff auf die Kamera, Kontakte, SD Karte, etc. Der App Store zeigt die benötigten *Capabilities* einer App an. Windows 10 Mobile fragt zur Laufzeit nach, ob gewisse *Capabilities* gewährt werden sollen oder nicht. Über die Einstellungen können die *Capabilities* jederzeit angepasst werden. Das Sandboxing-Konzept erlaubt eine absolute Isolation von am Gerät installierten Applikationen. Ein Zugriff auf den Speicherbereich, als auch auf Tastatureingaben einer anderen Applikation ist nicht möglich. Zusätzlich unterstützt Windows 10 Mobile Enterprise Data Protection (EDP). EDP hilft private Daten und Unternehmensdaten zu trennen und kann dadurch Datenlecks verhindern. Mittels MDM und EDP kann beispielsweise festgelegt werden welche Applikationen auf Firmendaten zugreifen dürfen.

Um die Integrität des Betriebssystems zu gewährleisten, verwendet Windows 10 Mobile Secure Boot und Windows Trusted Boot [3]. Wenn ein Gerät mit UEFI dem Nachfolger des BIOS startet und Secure Boot aktiviert ist wird die Integrität des Windows Bootloaders überprüft. Das System startet nur, wenn der Bootloader nicht modifiziert wurde (Signatur gültig) und vertrauenswürdig ist (vertrauenswürdiger Unterzeichner). Windows Trusted Boot überprüft dann die Windows Boot Komponenten inklusive der Microsoft Driver. Parallel dazu können mittels Measured Boot Informationen gesammelt und an einen Server geschickt werden, die beweisen, dass das System erfolgreich überprüft wurde. Durch die Verwendung dieser Technologien wird ein sehr guter Schutz gegen Bootkits und Rootkits erreicht.

Der für das Prüfen von Signaturen benötigte öffentliche Signaturschlüssel, der sogenannte *Platform-Key,* befindet sich in der UEFI-Umgebung, die für das Prüfen der Signaturen der Software-Komponenten zuständig ist. Dieser Schlüssel kann nach der Auslieferung des Geräts nur gelesen, aber nicht verändert oder gelöscht werden.

4.2.2. Verschlüsselung

Die Windows 10 Mobile Plattform bietet sowohl eine komplette Dateisystemverschlüsselung als auch ein applikationsspezifisches Verschlüsselungssystem. Die Dateisystemverschlüsselung basiert auf dem aus dem Desktopbereich bekannten *Bitlocker-System* [3]. Dabei kommt ein in der sicheren Umgebung eines Hardware-Elements abgelegter Schlüssel zum Einsatz. Die Verschlüsselung kann entweder manuell aktiviert werden oder per MDM-Regel erzwungen werden. Es wird der gesamte interne Speicher verschlüsselt, inklusive dem Betriebssystem und den

[12] Unter https://msdn.microsoft.com/en-us/library/windows/apps/mt270968.aspx ist eine Auflistung aller in Windows 10 Mobile zur Verfügung stehenden *Capabilities* zu finden, letzter Zugriff am 29.02.2016

Datenpartitionen. SD-Karten werden nicht verschlüsselt. Es wurden keine Informationen zur genauen Schlüsselableitung gefunden. Aus [3] [4] [5] geht jedoch hervor, dass der Schlüssel an die Hardware gebunden ist. Dadurch ist ein externer *Brute-Force-Angriff*, bei dem mithilfe von Rechen-Clustern eine Entschlüsselung mit allen möglichen Schlüssel versucht wird, praktisch nicht möglich. Der von der Benutzerin bzw. vom Benutzer gesetzte Passcode fließt nicht in die Schlüsselableitung mit ein. Bei der Durchführung eines *Jailbreaks*, könnte somit die Bildschirmsperre umgangen werden und das Gerät entsperrt werden.

Beim applikationsspezifischen Verschlüsselungssystem handelt es sich um die auf der Windows-Plattform vorhandene Data Protection API (DPAPI)[13]. Mithilfe der DPAPI können Entwicklerinnen und Entwickler einzelne Dateien im Speicherbereich der Applikation verschlüsseln. Der aus dem Passcode der Benutzerin bzw. des Benutzers abgeleitete Hashwert schützt einen zufällig von der DPAPI generierten Master Key. Um *Brute-Force-Angriffen* oder sogenannten *Rainbow-Table-Attacken* entgegen zu wirken, fließt zusätzlich ein Salt-Wert in die Schlüsselableitung mit ein. Werden Daten über die DPAPI verschlüsselt, wird aus dem Master Key, einem von der DPAPI generierten Zufallsstring und einer optionalen von der Entwicklerin bzw. dem Entwickler angegebenen Entropie ein Session Key abgeleitet, welcher die Daten verschlüsselt.

4.2.3. Zugriffsschutz

Windows 10 Mobile Geräte erlauben es der Benutzerin bzw. dem Benutzer einen Passcode zum Sperren des Gerätes festzulegen. Dabei sind Ziffernfolgen in der Länge von 4 - 127 Zeichen möglich. Über MDM Regeln kann die Stärke und eine Frequenz zur Änderung des Passcodes erzwungen werden. Alphanumerische Passcodes sind jedoch nur unter Verwendung von MDM-Regeln möglich. Zusätzlich kann man sich mittels Windows Hello[14] per Biometrie anmelden. Das Lumia 950 bzw. Lumia 950XL erlaubt beispielsweise die Anmeldung via Iris-Erkennung. Wie bereits in Abschnitt 4.2.2 erwähnt, hat der gewählte Passcode keinen Einfluss auf die Dateisystemverschlüsselung.

4.2.4. Applikationsquellen

Windows Mobile Applikationen werden von Microsoft kryptographisch signiert. Nicht von Microsoft signierte Applikationen können im Allgemeinen nicht am Gerät installiert werden. Somit steht Endbenutzerinnen und Endbenutzern der Windows Store bzw. der Windows Store for Business als alleinige Quelle für Applikationen zur Verfügung[15]. Applikationen werden vor ihrer Veröffentlichung von Microsoft auf ihre Funktionstüchtigkeit, Performance und Legitimität der von der Benutzerin bzw. vom Benutzer gesammelten Daten geprüft[16]. Da Sideloading von Applikationen unterbunden wird, können über in Phishing Mails oder in Kurznachrichten versandte Links keine bösartigen Applikationen installiert werden. Eine Ausnahme bilden bei Microsoft als Entwicklergeräte registrierte Smartphones, die es Entwicklerinnen und Entwicklern ermöglichen während der Implementierung von Applikationen diese am Gerät zu testen. Dazu wird jedoch ein kostenpflichtiges Entwicklerkonto bei Microsoft benötigt.

Um es auch Unternehmen zu ermöglichen, Unternehmensapplikationen an ihre Mitarbeiterinnen und Mitarbeiter zu verteilen, können Unternehmen den Windows Store for Business verwenden. Zu beachten ist, dass nur Unternehmensapplikationen installiert werden können, wenn auch von diesem Unternehmen ein Firmenkonto im Zuge eines MDM-Szenarios am Gerät installiert wurde.

---

[13] Ausführlichere Informationen zur Schlüsselhierarchie der DPAPI sind unter http://msdn.microsoft.com/en-us/library/ms995355.aspx zu finden. Letzter Zugriff: 10.04.2014

[14] http://windows.microsoft.com/de-at/windows-10/getstarted-what-is-hello

[15] Windows Store: https://www.microsoft.com/de-de/store/apps/windows-phone, letzter Zugriff: 29.02.2016

[16] Eine Liste der Anforderungen an Applikationen für ihre Veröffentlichung im Windows Phone Store ist unter https://msdn.microsoft.com/de-de/library/windows/apps/dn764944.aspx zu finden. Letzter Zugriff: 29.02.2016

4.2.5. Updatesituation

Microsoft garantiert für Windows 10 Mobile und Windows 10 Mobile Enterprise Updates bis zum 9.1.2018[17]. Es ist zu beachten, dass sich die tatsächlichen Updatezeiträume für Geräte mit Branding verkürzen können, falls sich Mobilfunkbetreiber dazu entscheiden, keine Updates weiterzugeben. Da Microsoft als alleinige Instanz das Betriebssystem wartet, besteht im Gegensatz zu Android bei Windows 10 Mobiler keine Möglichkeit für Mobilfunkbetreiber, Anpassungen am Betriebssystem vorzunehmen.

Aufgrund von Updateverzögerungen bei Geräten mit Branding empfiehlt sich vor allem bei Unternehmensgeräten der Kauf von Geräten ohne Branding, um zu gewährleisten, dass erkannte Sicherheitslücken ehestmöglich geschlossen werden können.

Generell wird die Updatesituation bei Windows 10 Mobile Geräten jedoch als sehr gut bewertet, da Microsoft als einziger Hersteller einen gewissen Updatezeitraum garantiert. Im Vergleich zu Android-Geräten ist bei Windows 10 Mobile Geräten des Weiteren mit keiner signifikanten Fragmentierung bezüglich der verwendeten Betriebssystemversionen und der angebotenen Funktionalität zu rechnen, da weder Gerätehersteller noch Mobilfunkbetreiber das Betriebssystem modifizieren.

4.2.6. Cloud-Anbindung

Im Gegensatz zu anderen Smartphone Plattformen verfügt Windows Mobile 10 über keine komplette Backuplösung, d.h. es ist nicht möglich, ein Backup des gesamten Dateisystems mit Cloud-Diensten zu synchronisieren. Jedoch können Nachrichten, Fotos, Inhalte von teilnehmenden Apps Systemeinstellungen sowie Konten und Kennwörter in der Cloud gesichert werden[18]. Microsoft verwendet hierfür OneDrive. Es ist Drittanbietern nicht möglich, eine komplette Backuplösung für Windows 10 Mobile Geräte zu implementieren, da herkömmliche Applikationen keinen Zugriff auf Daten anderer Applikationen haben.

4.2.7. MDM

Alle Windows 10 Plattformen (Desktop, Mobile und Internet of Things) kommen mit einem integrieren MDM-Client. Der Client unterstützt zwei wichtige Funktionen, Geräteregistrierung und Gerätemanagement. Neben Microsofts Azure Directory Service und Microsoft Intune gibt es noch einige Produkte von Drittanbietern die die Geräteregistrierung bzw. das Gerätemanagement unterstützen. Alle MDM-APIs sind offen und können von allen Hersteller verwendet werden. Neben Microsoft bieten beispielsweise AirWatch[19], Citrix[20], Lightspeed Systems[21], Matrix42[22], MobileIron[23], SAP[24], SOTI[25] und Symantec[26] entsprechende MDM Lösungen an. Mittels der MDM Systeme können beispielsweise Policies festgelegt werden, Applikationen installiert und aktualisiert werden oder andere Managementaufgaben durchgeführt werden. Die meisten Policies[27] sind sowohl für Windows 10 Mobile als auch Windows 10 Mobile Enterprise verfügbar. Zu den wichtigsten Fähigkeiten von Windows 10 Mobile Enterprise, die in Windows 10 Mobile fehlen, gehören die Möglichkeit, Software Updates aufzuschieben, mehr als 20 selbstsignierte Line-of-Business (LOB) Applikationen zu installieren und einzuschränken welche Telemetrie Daten gesammelt werden. Das Limit von 20 LOB Applikationen kann durch die Verwendung des Windows Store for Business oder den Kauf eines Code Signing Zertifikates von Verisign umgangen werden. Windows 10 Mobile

[17] https://support.microsoft.com/en-us/lifecycle/search?sort=PN&alpha=Windows%2010%20Mobile&Filter=FilterNO#;
[18] http://windows.microsoft.com/de-de/windows-10/getstarted-back-up-mobile
[19] http://www.air-watch.com/
[20] https://www.citrix.com/
[21] http://www.lightspeedsystems.com/
[22] https://www.matrix42.com/en/
[23] https://www.mobileiron.com/en
[24] http://go.sap.com/product/technology-platform/secure-mobile-device-management-cloud.html
[25] http://www.soti.net/
[26] https://www.symantec.com/
[27] https://technet.microsoft.com/en-us/library/dn904962(v=vs.85).aspx

unterstützt sowohl Firmentelefone als auch private Telefone. Es kann nur ein Konto verwendet werden um ein Gerät zu aktivieren. Für den BYOD Fall muss das Gerät mittels eines privaten Kontos (Microsoft Konto) aktiviert werden und anschließend muss das Gerät mit dem Azure-Konto (Firmen Konto) registriert werden. Firmentelefone müssen mittels Azure-Konto registriert werden. In diesem Fall erhält die Firma die komplette Kontrolle über das Gerät und kann die private Nutzung blockieren. Benutzer und Benutzerinnen können das Gerät nicht eigenständig der Kontrolle entziehen. Über Policies können unter anderem Einschränkungen bezüglich des verwendeten Passcodes (z.B. Passcode-Länge, alphanumerisch oder Sonderzeichen notwendig, Rotationsintervall) und der Benutzung der SD-Karte getroffen werden. Des Weiteren kann die Dateisystemverschlüsselung aktiviert werden. Im Gegensatz zu Windows Phone 8 kommt Windows 10 Mobile mit einem VPN Client und erlaubt beispielsweise immer eine VPN Verbindung zu erzwingen oder nur für bestimmte Applikationen. Weiteres kann beispielsweise nun die Kamera per Policy deaktiviert werden. Bezüglich Zertifikaten kann festgelegt werden ob der Benutzer bzw. die Benutzerin manuell Root und Zwischenzertifikate installieren darf.

4.2.8. BYOD

Wird das Gerät mittels eines privaten Kontos aktiviert und anschließend ein Azure AD Konto hinzugefügt wird das Gerät im BYOD-Modus betrieben. Die Unterschiede zwischen einem privaten und einem Firmengerät sind in Tabelle 1 zusammengefasst.

	Privatgerät	Firmengerät
Aktivierung	Microsoft Konto	Azure AD Konto
Einloggen	Das Gerät kann nicht mittels Azure AD Konto entsperrt werden.	Benutzer und Benutzerinnen können das Gerät mittels Azure AD Konto entsperren. Das Hinzufügen von privaten Konten kann verboten werden.
Synchronisation von Benutzer und Applikationseinstellungen über mehrere Geräte	Die Benutzereinstellungen und Applikationseinstellungen werden zwischen allen Geräten mit demselben Microsoft Konto über das private OneDrive-Konto synchronisiert.	Windows 10 Mobile unterstützt derzeit nicht die Synchronisation der Benutzereinstellungen und Applikationseinstellungen über die Firmencloud. Die Synchronisation über die private Cloud kann deaktiviert werden.
Möglichkeit private Konten zu sperren	Nein	Ja
Kontrollniveau	Firmen können die meisten Regeln anwenden (ändert sich möglicherweise in der Zukunft), können aber nicht das Microsoft Konto entfernen. Benutzer und Benutzerinnen des Gerätes können jederzeit die volle Kontrolle über ihr Gerät zurück erlangen indem Sie das Gerät der MDM-Kontrolle entziehen.	Es können alle unterstützen Regeln verwendet werden. Benutzer oder Benutzerinnen können das Gerät nicht der MDM Kontrolle entziehen.

Tabelle 1: Unterschiede bei der MDM Kontrolle zwischen Privatgeräten und Firmengeräten

4.3. BlackBerry 10

Letzte Aktualisierung	April 2014

Um auf die Anforderungen von BYOD zu reagieren, brachte BlackBerry 2013 mit BlackBerry 10 (BB10) ein stark überarbeitetes Betriebssystem auf den Markt. Im nachfolgenden Abschnitt werden die von BlackBerry 10 angebotenen Sicherheitsfunktionen beschrieben und vor allem *BlackBerry Balance*, die im Betriebssystem integrierte BYOD-Lösung näher betrachtet. Da im Vergleich zu Android oder iOS kaum tiefgehende Analysen der Sicherheit der Plattform oder bestimmter Mechanismen verfügbar sind, kann an manchen Stellen nur eine oberflächliche Betrachtung geboten werden.

4.3.1. Basissicherheit

BB10 besteht aus einem QNX Microkernel [6]. Microkernel bieten den Vorteil, dass nur der Kernel und Komponenten zur Verwaltung von Prozessen im privilegierten Kernelmodus laufen. Alle Gerätetreiber, Applikationen, etc. laufen in einem eigenen Prozess mit eigener UID und können nur auf den ihnen zugewiesenen Speicherbereich zugreifen. Wie auch iOS, Windows 10 Mobile und Android unterstützt BB10 ASLR und DEP [6], um es Angreifern nicht zu ermöglichen, eingeschleusten Code auszuführen, bzw. um die Vorhersage von Speicheradressen bestimmter Programmfragmente zu erschweren.

Bei BB10 Geräten wird im Zuge des Bootvorgangs sowohl die Signatur des geladenen Boot ROM, des Betriebssystems als auch aller Applikationen am Gerät verifiziert [6]. Laut BlackBerry-Dokumentation sind im Applikationsmanifest einer Applikation signierte Hashwerte von Applikationskomponenten abgelegt. Bei der Installation oder der Aktualisierung einer Applikation werden die Hashwerte und Signaturen geprüft. Damit kann sichergestellt werden, dass auch bei unverschlüsselter Übertragung der Softwarekomponenten über die Netzwerkschnittstelle diese nicht modifiziert wurden.

Beim ersten Öffnen einer Applikation wird die Benutzerin bzw. der Benutzer um ihr bzw. sein Einverständnis bezüglich der von der Applikation angeforderten Permissions gebeten. Im Gegensatz zu älteren Android Versionen können selektiv Permissions erteilt werden[28] und auch nachträglich erteilt bzw. entfernt werden. Als zusätzlicher Sicherheitsgewinn im Vergleich zu Android kann die geringe Anzahl an Permissions[29] und deren niedrigere Granularität genannt werden.

4.3.2. Verschlüsselung

BB10 bietet eine Trennung von persönlichen und geschäftlichen Daten auf Dateisystemebene. Um sowohl persönliche als auch geschäftliche Daten im Falle eines Diebstahles vor unerlaubten Zugriffen zu schützen, verfügt BB10 über zwei unterschiedliche Verschlüsselungssysteme [6]. Zum einen verfügt BB10 über eine standardmäßig aktive Verschlüsselung des *Work Space* und zum anderen über eine Verschlüsselung des *Personal Space*, die manuell von der Benutzerin bzw. dem Benutzer aktiviert werden muss oder über MDM-Regeln gefordert werden kann. Zusätzlich ist eine Verschlüsselung der SD-Karte möglich. Es ist jedoch anzumerken, dass Work-Space-Daten niemals auf der SD-Karte abgelegt werden.

Abbildung 4 beschreibt die bei BB10 verwendete Schlüsselhierarchie bei der Verschlüsselung des *Work Space*. Für jede zu verschlüsselnde Datei wird ein zufälliger *File Encryption Key* erzeugt. Der jeweilige *Domain Key* (*Work Domain Key* oder *Personal Domain Key*) schützt alle *File Encryption Keys* der jeweiligen Domäne. Sowohl der *Domain Key* als auch die *File Encryption Keys* werden im Dateisystem abgelegt. Die *Domain Keys* werden über den im NVRAM (*Non Volatile Random Access Memory*, dt. nicht flüchtiger Arbeitsspeicher) gehaltenen *Work Master Key* bzw. *Personal Master*

[28] https://labs.mwrinfosecurity.com/system/assets/410/original/mwri_blackberry-10-security_2013-06-03.pdf , letzter Zugriff am 14.04.2014

[29] Liste aller unter BB10 verfügbaren Permissions: https://developer.blackberry.com/air/documentation/accessing_secure_apis_1524628_11.html, letzter Zugriff am 14.04.20114

Key geschützt, der wiederum vom *System Master Key* verschlüsselt wird. Der *System Master Key*, der beide *Domain Master Keys* verschlüsselt, wird von einem bei der Herstellung des Geräts im Prozessor abgelegten zufälligen Schlüssel geschützt.

Der von der Benutzerin bzw. vom Benutzer gewählte Passcode fließt weder im *Work Space* noch im *Personal Space* in die Ableitung der kryptographischen Schlüssel mit ein. Daraus ergibt sich ähnlich wie bei iOS und Windows Phone 8, dass im Falle eines *Jailbreaks* für BB10, der die Bildschirmsperre umgeht, die Daten am Gerät nicht mehr geschützt sind. Es ist jedoch anzumerken, dass zum momentanen Zeitpunkt noch kein *Jailbreak* für BB10 bekannt wurde.

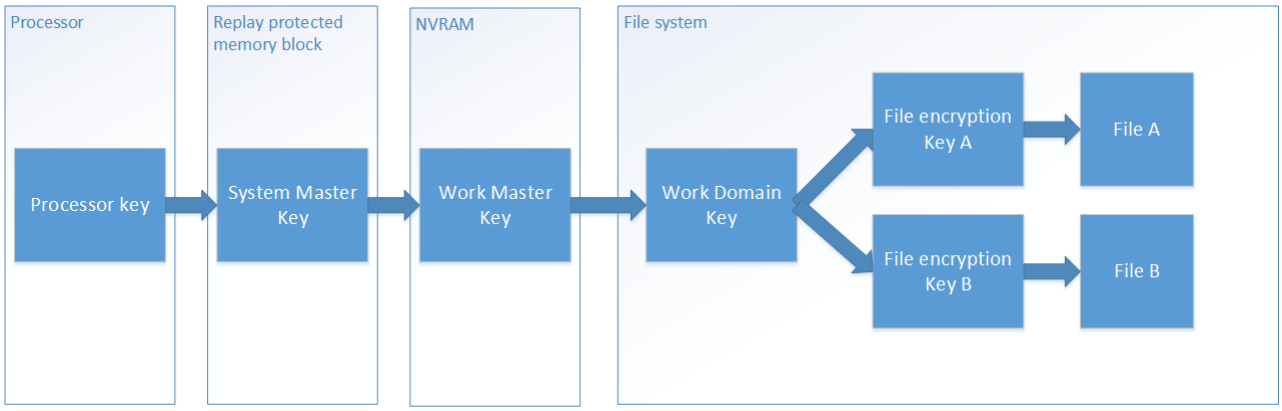

Abbildung 4: BB10 Schlüsselhierarchie Work Space

4.3.3. Zugriffsschutz

Bei BB10 kommen zwei unterschiedliche Passcodes zur Anwendung. Zum einen der verpflichtende Passcode, um den *Work Space*, der die Unternehmensdaten beherbergt, zu schützen und zum anderen der optionale Passcode für den *Personal Space* der Benutzerin bzw. des Benutzers. Bei aktivierter Verschlüsselung im *Personal Space* kann über die MDM-Regel *Apply Work Space Password to Full Device* eine Ausweitung des *Work-Space*-Passcodes auf das gesamte Gerät ermöglicht werden, um der doppelten Eingabe von Passcodes entgegenzuwirken.

Bei zehnmaliger falscher Passcode-Eingabe werden alle Daten am Gerät gelöscht. Benutzerinnen und Benutzer können die Anzahl falscher Eingabeversuche jedoch erhöhen [6].

4.3.4. Applikationsquellen

BlackBerry World[30] stellt die offizielle Quelle von Applikationen für BlackBerry-Geräte dar. BlackBerry ermöglicht im Gegensatz zu iOS und Windows Phone 8 die Installation von Applikationen aus alternativen Quellen („Sideloading“). Dies gilt sowohl für BlackBerry-Applikationen als auch für die Installation von Android-Applikationen. Bis zur Version 10.2 erforderte die Installation von Android-Applikationen eine manuelle Konvertierung der von Android verwendeten *APK-Dateien* zu BlackBerry *BAR*-Dateien. Seit Version 10.2 kann der Amazon-Applikationsstore am BlackBerry-Gerät installiert werden und Android Applikationen über diesen direkt bezogen werden[31].

Für den *Work Space* existiert eine eigene Version von *BlackBerry World*, die *BlackBerry World for Work*. Diese beinhaltet sowohl vom jeweiligen Unternehmen betriebene Applikationen als auch Applikationen von Drittanbietern. Benutzerinnen und Benutzer können allerdings nur vom Unternehmen genehmigte Applikationen von Drittanbietern im *Work Space* installieren. Sowohl Applikationen der öffentlichen *BlackBerry World* als auch Android Applikationen können nur im persönlichen Bereich des Geräts installiert werden.

[30] Blackberry App World: https://appworld.blackberry.com/ , letzter Zugriff am 14.04.2014
[31] Installation von Android-Applikationen: http://www.cnet.com/how-to/the-easiest-way-to-install-android-apps-on-bb10/ , letzter Zugriff am 15.04.2014

4.3.5. Updatesituation

Seit der Veröffentlichung der Version 10 im März 2013 wurden bereits zwei große Versionsupdates veröffentlicht[32]. Mit den Versionen 10.1 und 10.2 wurde neue Funktionalität eingeführt. Zusätzlichen wurde eine Vielzahl an kleinen Updates mit Bugfixes verteilt. Die Häufigkeit der Updates im ersten Erscheinungsjahr von BB10 lässt auf eine gute Updatesituation schließen.

4.3.6. Cloud-Anbindung

Laut den verfügbaren Informationen bietet BlackBerry keinen zentralen Cloud-Backup-Dienst an. Unverschlüsselte Backups können jedoch von der Benutzerin bzw. vom Benutzer über die Software BlackBerry Link am Desktop-Computer angelegt werden[33].

4.3.7. MDM

Mit *BlackBerry Balance* gelingt BlackBerry eine komplette Trennung des geschäftlichen Bereichs vom privaten Bereich. Über MDM-Regeln kann sowohl der private als auch der geschäftliche Teil geregelt werden. Für den privaten Bereich stehen allerdings nur eingeschränkte Regeln zur Verfügung. Diese Variante stellt sowohl für MDM als auch BYOD die beste Lösung dar, da die Trennung von privaten und geschäftlichen Daten bereits im Betriebssystem integriert ist und somit größte Sicherheit und Verwendbarkeit für die Endbenutzerin bzw. den Endbenutzer geboten werden kann. Abbildung 5 illustriert das Konzept der Trennung von privatem und geschäftlichem Bereich.

Bei der Aktivierung des Geräts über das *BlackBerry Device Service* wird der geschäftliche Bereich am Gerät erstmals angelegt. Die Aktivierung hat keine Auswirkungen auf die am Gerät befindlichen privaten Daten. Für die beiden Bereiche stehen unterschiedliche Applikations-Stores zur Verfügung. Im geschäftlichen Bereich können Applikationen über die *BlackBerry World for Work* bezogen werden, die ausschließlich aus vom jeweiligen Unternehmen genehmigten Applikationen besteht.

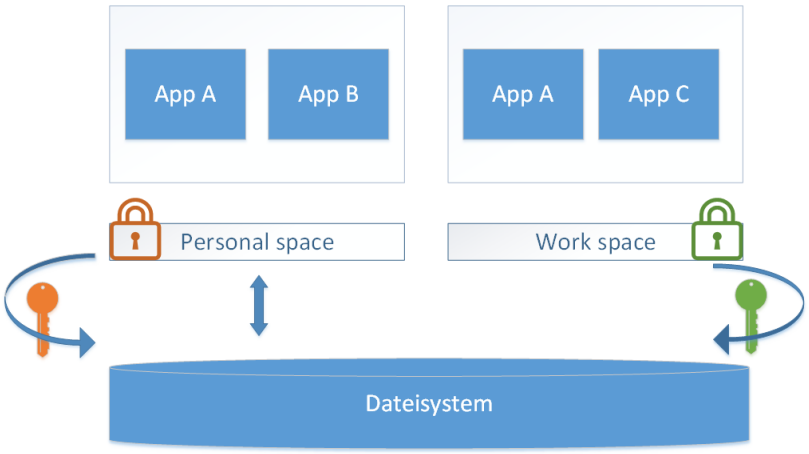

Abbildung 5: Konzeptueller Aufbau von BlackBerry Balance [6]

Eine Auflistung der unter BB10 verfügbaren MDM-Regeln [6]:

- **Zugriffsschutz**
  - Apply Work Space Password to Full Device
  - Maximum Password Age
  - Maximum Password Attempts
  - Minimum Password Complexity

[32] http://en.wikipedia.org/wiki/BlackBerry_10#Version_history , letzter Zugriff am 15.04.2014

[33] Blackberry Link: http://us.blackberry.com/software/desktop/blackberry-link.html, letzter Zugriff am 15.04.2014
Blackberry Link verfügt über keine Möglichkeit zur Verschlüsselung von Backups:
http://supportforums.blackberry.com/t5/BlackBerry-Link/Encrypting-backup-with-Link-on-Mac/td-p/2643509 , letzter Zugriff am 15.04.2014

    - Maximum Password History
    - Minimum Password Length
    - Password Required for Work Space
    - Security Timeout
- **Sicherheit**
    - Backup and Restore Work Space
    - Media Card Encryption
    - Network Access Control for Work Apps
    - Personal Apps Access to Work Contacts
    - Personal Space Data Encryption
    - Share Work Data During BBM Video Screen Sharing
    - Voice Control
    - Voice Dictation in Work Apps
    - Wipe the Work Space Without Network Connectivity
    - Work Network Usage for Personal Apps
- **Software**
    - BBM Video Access to Work Network
    - Cloud Storage Access from Work Space
    - Open Links in Work Email Messages in the Personal Browser
    - Transfer Work Contacts Using Bluetooth PBAP or HFP
    - Transfer Work Files Using Bluetooth OPP
    - Transfer Work Messages Using Bluetooth MAP

Benutzerinnen und Benutzern ist es nicht möglich, Dateien zwischen *Work Space* und *Personal Space* zu verschieben. Über die MDM-Regeln *Personal Apps Access to Work Contacts* kann jedoch im persönlichen Bereich ein Zugang zu geschäftlichen Kontakten erreicht werden. Sollten im geschäftlichen Bereich jedoch beim Verfassen einer E-Mail private Kontakte angegeben werden, so wird die Benutzerin bzw. der Benutzer vom System darauf hingewiesen. Umgekehrt ist es möglich, z.B. private Dateien als Anhänge geschäftlicher E-Mails zu senden, jedoch erhält der geschäftliche Bereich dabei lediglich Lesezugriff auf die Daten des persönlichen Bereichs.

4.3.8. BYOD

BB10 bietet als bisher einziger Hersteller von Smartphone-Plattformen eine im Betriebssystem integrierte BYOD Lösung. Sicherheitsmechanismen wie Remote Wipe sind ausschließlich auf den geschäftlichen Bereich anwendbar. Es ergibt sich bezüglich Verwaltung von BlackBerry-Geräten und angebotener Funktionalität keine Unterscheidung zwischen MDM-Szenarien und BYOD.

4.4. Android

Analysierte Version	Android 6 (Marshmallow)
Letze Aktualisierung	Mai 2016

Mit einem weltweiten Marktanteil von ca. 85% im dritten Quartal des Jahres 2015, bei verkauften Geräten[34], hat sich die 2008 auf den Markt gekommene Android-Plattform zum meistgenutzten Smartphone-Betriebssystem entwickelt. Android wird hauptsächlich von Google als Open-Source-Projekt betrieben und von mehreren Smartphone-Herstellern als mobiles Betriebssystem genutzt. Da Android im Vergleich zu iOS oder Windows 10 Mobile eine eher offene Applikationspolitik vertritt und Applikationen auch aus alternativen Quellen installiert werden können, ergeben sich hohe Anforderungen an die Sicherheit der Plattform. Dieser Abschnitt bieten einen Überblick über die unter Android zur Verfügung stehenden Sicherheitsfunktionen.

4.4.1. Basissicherheit

Android verfügt seit der Version 4.0 über ASLR zur Randomisierung von Speicherbereichen[35]. Erst seit Android 4.1 wird ASLR in vollem Umfang unterstützt, zuvor waren Attacken bei denen zwar kein neuer Code eingeschleust wurde, jedoch vorhandener Code aufgerufen wurde (sogenannte *Return-Oriented-Programming* Angriffe) möglich. Seit Version 2.3 nutzt Android *Data Execution Prevention* (DEP), um Daten als nicht ausführbar zu markieren. DEP ist standardmäßig aktiv und kann nicht deaktiviert werden.

Seit Android 4.3 wird auch SELinux (Security-Exhanced Linux) unterstützt. Damals lief nur ein kleiner Teil des Systems (*installd, netd, vold* und *zygote*) im „*enforcing*"-Modus. Andere Teile liefen im „*permissive*"-Modus, wo Zugriffsverweigerungen nur aufgezeichnet aber nicht durchgesetzt wurden. Seit Android 5 läuft alles im „*enforcing*"-Modus. SELinux limitiert im Falle eines erfolgreichen Angriffs (z.B. über einen Buffer Overflows) die verfügbaren Zugriffsmöglichkeiten und kann daher die Auswirkungen des Angriffs einschränken.

Jede Applikation verfügt über einen eigenen isolierten Speicherbereich, der vor Zugriffen anderer Applikationen geschützt ist. Ein Zugriff auf diesen Speicherbereich ist lediglich Applikationen derselben Entwicklerin bzw. desselben Entwicklers möglich. Android realisiert diesen Mechanismus über die Verwendung von Linux-Prozessen, wonach jede Applikation mit einer eigenen User-ID in einem eigenen Prozess ausgeführt wird. Die Kommunikation zwischen Applikationen erfolgt über definierte Schnittstellen. Dabei können Daten von einer Applikation mithilfe sogenannter *Intents* an andere Applikationen weitergeleitet werden. Falls nicht von der Entwicklerin bzw. vom Entwickler unterbunden[36], besteht zusätzlich die Möglichkeit auf *Services* anderer Applikationen zuzugreifen. Ein *Service* bezeichnet eine Hintergrundoperation, die u.a. für die Kommunikation mit einem Webserver oder das Abspielen von Musik verwendet werden kann.

Standardmäßig verfügen Android Applikationen weder über die Möglichkeiten, Daten der Benutzerin bzw. des Benutzers (z.B. Kontakte, Nachrichten) auszulesen noch Funktionen des Geräts (z.B. Kamera oder Bluetooth) anzusprechen. Applikationsentwicklerinnen und Applikationsentwickler müssen im Manifest der Applikation sogenannte *Permissions* angeben, um Zugriff auf diese Funktionen oder Daten zu erlangen. Bei der Installation einer Applikation wird die Benutzerin bzw. der Benutzer um Einverständnis gebeten, ob der Applikation die eingetragenen Rechte zukommen sollen. Bis einschließlich Android 5 wurde hierbei ein „*all-or-nothing*"-Ansatz verfolgt, wobei eine Installation der Applikation nur bei Bestätigung aller geforderten *Permissions* möglich war. Seit Android 6 werden sogenannte *Runtime Permissions* unterstützt. Dadurch können Permissions zur Laufzeit bewilligt bzw. entzogen werden. Android verfügt über eine Vielzahl an teilweise sehr fein

[34] Gartner, Smartphone Verkäufe Q3 2015 http://www.gartner.com/newsroom/id/3169417 , letzter Zugriff am 08.03.2016

[35] Offizielle Webseite von Google bezüglich Plattformsicherheit: https://source.android.com/security/ , letzter Zugriff am 18.05.2016

[36] Um einen Zugriff auf Komponenten durch andere Applikationen zu unterbinden muss im Manifest der Applikation das Attribut „android:exported" der jeweiligen Komponente auf „false" gesetzt werden. Siehe http://developer.android.com/guide/topics/manifest/service-element.html, letzter Zugriff am 14.04.2014

granulierten *Permissions*[37], die es Entwicklerinnen und Entwickler erschweren den Überblick zu behalten und nur tatsächlich verwendete *Permissions* zu deklarieren. Des Weiteren beinhalten die bei der Installation geforderten *Permissions* zwar eine kurze Beschreibung des geforderten Funktionsumfangs, jedoch ist vor allem für unerfahrene Benutzerinnen und Benutzer oftmals unverständlich, welche Konsequenzen die Bestätigung dieser nach sich ziehen kann. Durch die *Runtime Permissi*ons verbessert sich die Situation, da nun direkt vor dem Zugriff auf die Ressource gefragt wird.

Die im Herbst 2013 veröffentlichte Android Version 4.4 verfügt über eine experimentelle Verifizierung der im Zuge des Bootvorgangs geladenen Komponenten[38]. Google nennt dieses Sicherheitsfeature *Verified Boot*. Die dazu verwendete Kernelfunktionalität *dm-verity* muss jedoch vom Gerätehersteller unterstützt und konfiguriert werden. *dm-verity* überprüft die Integrität der Speicherblöcke auf der Festplatte, um so sicherzustellen, dass sich das Gerät nach einem Reboot im bei der letzten Verwendung hinterlassenen Zustand befindet. Durch die Überprüfung der Hash-Werte aller Speicherblöcke können persistente Rootkits detektiert werden. Durch die Verwendung von *dm-verity* wird die Installation von modifizierter Software (z.B. Custom ROMs) erheblich erschwert, da diese oftmals Root-Zugriff voraussetzen [7]. Seit Android 6 ist *Verified Boot* standardmäßig aktiviert und garantiert die Echtheit aller Komponenten vom Bootloader bis zum Betriebssystem.

Im November 2014 hat Google Android 5 veröffentlicht. Auch Android 5 brachte einige neue und verbesserte Sicherheitsfeatures [8]. Seit Android 5 wird beispielsweise das Dateisystem standardmäßig verschlüsselt. Ebenfalls seit Android 5 wird nun TLS in der Version 1.1 und 1.2 unterstützt. Ein weiteres wichtiges Sicherheitsfeature sind die direkten Updates für die *WebView*. Zuvor konnte die *WebView* nur im Rahmen eines Systemupdates aktualisiert werden. Ebenso wurde mit Android 5 das Multi-Benutzerkonzept für Telefone eingeführt. Android 4.2 hatte bereits Multibenutzerunterstützung für Tablets. Dadurch können mehrere Benutzerinnen bzw. Benutzer dasselbe Telefon oder Tablet benutzen, jede bzw. jeder mit eigenen Apps. Für kurze Zugriffe auf das Gerät, ohne Zugriff auf Daten und Apps, wurde ein Gast-Modus hinzugefügt. Speziell für Android for Work spielt das Multi-Benutzerkonzept eine wichtige Rolle.

Auch Android 6 brachte wieder einige Verbesserungen mit sich [9]. Neben den *Runtime Permissions* wurde ein *Hardware-Abstraktions-Layer (HAL)* eingeführt. Dieser wird von der Fingerabdruck API, vom Sperrbildschirm, von der Dateisystemverschlüsselung und für Client Zertifikate verwendet um das Schlüsselmaterial gegen Software- und Hardware-Attacken zu schützen. Zusätzlich wurde Android 6 mit weiteren detaillierten SELinux-Regeln gehärtet. Eine weitere wichtige Änderung gibt es beim USB Zugriff. Dieser wird standardmäßig nun im Lademodus betrieben. Wird der Zugriff auf Daten oder andere Funktionalität benötigt muss dieser explizit freigeben werden.

4.4.2. Verschlüsselung

Seit der Version 3.0 bzw. 4.0 für Smartphones verfügen Android-Geräte über die Möglichkeit der Dateisystemverschlüsselung[39]. Diese war jedoch nicht standardmäßig aktiv, sondern musste von der Benutzerin bzw. vom Benutzer manuell aktiviert oder über MDM-Regeln erzwungen werden. Seit Android 5 werden neue Geräte standardmäßig mit aktiver Verschlüsslung ausgeliefert[40] [10]. Falls ein Hardware KeyStore verfügbar ist, wird der Schlüssel ans Gerät gebunden.

Es wird in weiterer Folge das Verschlüsselungssystem von Android 6 beschrieben, Unterschiede zu früheren Android Versionen werden weiter unten diskutiert. Google gibt als Verschlüsselungsalgorithmus AES-128 im CBC Modus mit ESSIV:SHA256 vor. Beim ersten Start

[37] Liste aller unter Android verfügbaren Permissions: http://developer.android.com/reference/android/Manifest.permission.html, letzter Zugriff am 08.03.2016

[38] Verifizierter Boot-Vorgang mit dm-verity: https://source.android.com/security/verifiedboot/index.html, letzter Zugriff am 18.05.2016

[39] Detaillierte Informationen zur Implementierung der Dateisystemverschlüsselung: https://source.android.com/devices/tech/encryption/android_crypto_implementation.html , letzter Zugriff am 14.04.2014

[40] Ausgenommen sind nur schwächere Geräte, welche AES-Berechnungen mit einem Durchsatz von mindestens 52 Mbit/s nicht durchführen können.

generiert das Gerät einen zufälligen 128-Bit Disk Encryption Key (DEK) und einen 128-Bit Salt. Auf das beim ersten Start vorgegebenen Passwort und den generierten Salt wird die Schlüsselableitungsfunktion *scrypt* angewendet. Der resultierende Schlüssel IK1, wird auf die Größe des privaten Hardwareschlüssels (HBK) aufgefüllt. Mittels des HBKs wird der aufgefüllte IK1 signiert, um einen 256-Byte langen Schlüssel IK2 zu erstellen. Auf IK2 und den Salt wird *scrypt* angewendet um IK3 zu erstellen. Abschließend wird der DEK mit AES verschlüsselt, wobei die ersten 16 Byte von IK3 als Verschlüsselungsschlüssel und die letzten 16-Byte von IK3 als Initialisierungsvektor verwendet werden. Der verschlüsselte DEK und der Salt werden zusammen mit anderen wichtigen Parametern in den Krypto-Metadaten gespeichert. Die Informationen in den Krypto-Metadaten werden für zukünftige Startvorgänge benötigt. Abbildung 6 visualisiert das Verschlüsselungssystem von Android 6.

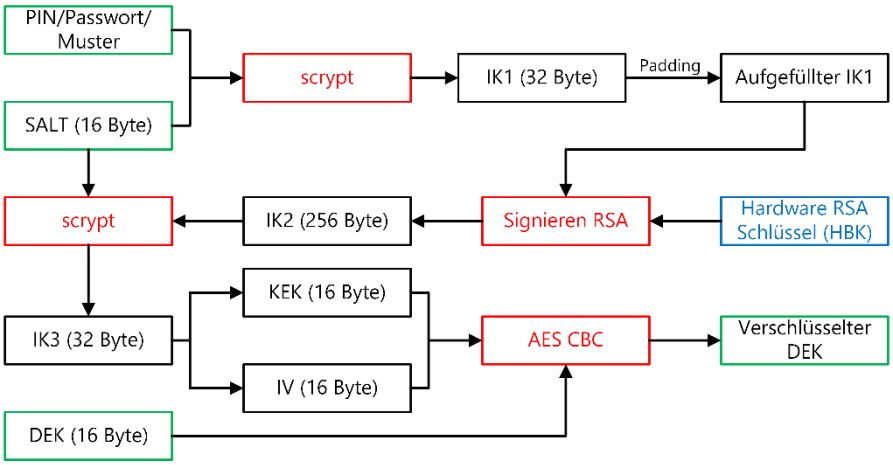

Abbildung 6: Verschlüsselungssystem von Android 6

Setzt die Benutzerin bzw. der Benutzer ein Password bzw. einen PIN muss nur der DEK neu verschlüsselt werden, wodurch die Neuverschlüsselung des gesamten Datenträgers vermieden werden kann. Wenn die Benutzerin bzw. der Benutzer kein Passwort, PIN oder Muster verwendet, wird ein vorgegebenes Passwort verwendet. Zusammenfassend kann gesagt werden, dass mit dieser Version des Verschlüsselungssystems ein etwaiges vorhandenes Hardware-Element in den Entschlüsselungsvorgang miteingebunden wird und somit Brute-Force Angriffe bei einem gestohlenen Gerät signifikant erschwert werden.

Bis Android 4.4 wurde das Hardware-Element nicht in den Verschlüsselungsprozess einbezogen, wodurch Brute-Force-Angriffe relativ effektiv auf externen Rechnern durchführbar waren. Vor Android 4.4 waren Brute-Force-Angriffe durch die Verwendung der Passwortableitungsfunktion PBKDF2 noch effektiver. Die Publikation von Teufl et al. bietet zusätzlich zu detaillierten Informationen zur Ableitung des Schlüssels und der dabei verwendeten Algorithmen [11] auch eine Auflistung von Dauer und Kosten für Brute-Force Angriffe in der Amazon EC2 Cloud.

Tabelle 2 zeigt die Brute-Force Zeiten für Android Versionen kleiner 4.4. Ersichtlich ist dabei, dass über eine vergleichsweise günstige Anmietung von Cloud-Instanzen z.B. selbst 8-stellige numerische Passcodes und alphanumerische Passcodes geringer Länge innerhalb kürzester Zeit kompromittiert werden können.

	Länge	***Mögliche Zeichen***	***Anzahl möglicher Passcodes***	***Dauer mit einer Instanz (in Tagen)***	***Dauer mit 1000 Instanzen (in Tagen)***
Numerisch	4	10	10.000	0,0	0,0
	6	10	1.000.000	0,2	0,0
	8	10	100.000.000	20,3	0,0
	10	10	10.000.000.000	2.025,5	2,0
Alphanumerisch	4	36	1.679.616	0,3	0,0
	6	36	2.176.782.336	440,9	0,4

(Kleinbuchstaben, Zahlen)	8	36	2,82111E+12	571.405,4	571,4
Alphanumerisch *(Groß- und Klein-buchstaben gemischt, Zahlen)*	4 6 8	62 62 62	14.776.336 56.800.235.584 2,1834E+14	3,0 11.504,7 44.223.979,7	0,0 11,5 44.224,0
Komplex *(Groß- und Klein-buchstaben gemischt, Zahlen, Sonderzeichen)*	4 6	107 107	131.079.601 1,50073E+12	26,5 303.967,4	0,0 304,0

Tabelle 2: Android < 4.4: Benötigte Zeiten für Brute-Force Angriffe (Daten entnommen von [11])

Durch den Wechsel der Schlüsselableitungsfunktion bei Android 4.4 auf *scrypt* steigt die benötigte Brute-Force Zeit signifikant an. Tabelle 3 bietet eine Übersicht der Dauer für Brute-Force Angriffe bei Geräten mit Android Version 4.4. Es ist anzumerken, dass Gerätehersteller die von scrypt verwendeten Parameter konfigurieren können, wodurch sich Unterschiede zwischen den Brute-Force Zeiten der verschiedenen Geräte ergeben können.

	Länge	***Mögliche Zeichen***	***Anzahl möglicher Passcodes***	***Dauer mit einer Instanz (in Tagen)***	***Dauer mit 1000 Instanzen (in Tagen)***
Numerisch	4 6 8 10	10 10 10 10	10.000 1.000.000 100.000.000 10.000.000.000	0,1 8,1 810,2 81.018,5	0,0 0,0 0,8 81,0
Alphanumerisch *(Kleinbuchstaben, Zahlen)*	4 6 8	36 36 36	1.679.616 2.176.782.336 2,82111E+12	13,6 17.636,0 22.856.214,5	0,0 17,6 22.856,2
Alphanumerisch *(Groß- und Klein-buchstaben gemischt, Zahlen)*	4 6 8	62 62 62	14.776.336 56.800.235.584 2,1834E+14	110,7 460.187,1 1.768.959.188,8	0,1 460,2 1.768.959,2
Komplex *(Groß- und Klein-buchstaben gemischt, Zahlen, Sonderzeichen)*	4 6	107 107	131.079.601 1,50073E+12	1.062,0 12.158.695,0	1,1 12.158,7

Tabelle 3: Android 4.4: Benötigte Zeiten für Brute-Force Angriffe (Daten entnommen von [11])

Seit Android 4.0 steht Applikationen eine KeyChain zur Speicherung von Passwörtern, privaten Schlüsseln und Zertifikaten zur Verfügung. Diese kann selbst bei nicht aktiver Dateisystemverschlüsselung verwendet werden. Die KeyChain wird über einen ebenfalls vom Passcode abgeleiteten Schlüssel geschützt, dieser verschlüsselt und entschlüsselt KeyChain-Einträge. Jeder KeyChain-Eintrag ist über die Applikations-ID einer Applikation eindeutig zugeordnet. Somit können Applikationen nicht auf KeyChain-Einträge fremder Applikationen zugreifen. Mit der Android Version 4.1 wurde die KeyChain API um Hardware-Unterstützung erweitert[41]. Dabei werden die KeyChain-Einträge zusätzlich über einen nur in der sicheren Umgebung des Hardware-Elements verfügbaren Schlüssel geschützt. Die Verfügbarkeit eines Hardware-Elements ist jedoch geräteabhängig.

4.4.3. Zugriffsschutz

Um das Gerät vor unerlaubten Zugriffen zu schützen, bietet Android mehrere Möglichkeiten des Zugriffsschutzes. Zusätzlich zur Wahl von PINs und Passcodes, bietet Android Verfahren wie *Pattern Unlock*, *Face Unlock oder Smart Lock*. Bei *Pattern Unlock* wird anstelle eines Passcodes zum Entsperren ein bestimmtes Muster am Bildschirm eingegeben. *Face Unlock* entsperrt das

[41] Android 4.1 Hardware-Sicherung für KeyChain-Schlüssel: http://developer.android.com/about/versions/jelly-bean.html, letzter Zugriff am 18.04.2014

Gerät, sobald das Gesicht der legitimen Benutzerin bzw. des legitimen Benutzers erkannt wird. Beide Verfahren gelten im Vergleich zu Passcodes als unsicher und sollten nicht verwendet werden, falls sich kritische Daten am Gerät befinden[42]. Android 5 unterstützt *Smart Lock*, dadurch kann das Gerät automatisch entsperrt werden, wenn es sich an einem vertrauenswürdigen Ort befindet, wenn vertrauenswürdige Bluetooth- oder NFC-Geräte in der Nähe sind, wenn ein vertrauenswürdiges Gesicht erkannt wird, oder wenn die Trageerkennung den Besitzer erkennt. Auch Smart Lock bietet nicht dasselbe Sicherheitslevel wie ein PIN oder ein Passcode. Google warnt etwa, dass die Trageerkennung möglicherweise nicht immer korrekt bestimmen kann, wer das Gerät gerade trägt[43]. Auch beim Entsperren an vertrauenswürdigen Orten sollte beachtet werden, dass das Gerät möglicherweise in einem Radius von bis zu 80 Metern um den definierten Ort entsperrt bleibt.

4.4.4. Applikationsquellen

Der Google Play Store[44] stellt die erste Anlaufstelle für Android Applikationen dar. Über eine am Gerät vorinstallierte Applikation können von der Benutzerin bzw. vom Benutzer passende Applikationen gesucht und installiert werden. Die Veröffentlichung von selbst entwickelten Applikationen erweist sich bei Android als vergleichsweise einfach, da Google auf einen umfassenden Review-Prozess verzichtet. Applikationen werden zwar automatisiert mittels Bouncer überprüft, es rutschen aber immer wieder bösartige Applikationen durch[45]. Im Vergleich zu anderen Applikationsquellen bietet Google Play aber ein hohes Sicherheitsniveau. Seit Android 4.2 können auch Applikationen aus unbekannten Quellen überprüft werden.

4.4.5. Updatesituation

Android wird von unzähligen Geräteherstellern als Betriebssystem für Smartphones, Tablets, mittlerweile auch *Smartwatches* und andere mobile Geräte genutzt. Die Gerätehersteller nehmen in vielen Fällen Änderungen an der Benutzeroberfläche vor und fügen neue Funktionalitäten hinzu. Die von den Herstellern angepassten Systeme werden oftmals nicht oder nur über einen kurzen Zeitraum mit Updates versorgt, da sich die Anpassungen an neue von Google veröffentlichte Versionen mitunter sehr zeitintensiv gestalten. Laut Google liefen Anfang Februar 2016 nur 1,2% der Geräte mit Android 6 (Marshmallow), welches Anfang Oktober des Vorjahres veröffentlicht wurde[46]. Die bei Android vorzufindende Fragmentierung hat sicherheitskritische Auswirkungen, da Sicherheitslücken bei Geräten mit alten Versionen nicht geschlossen wurden und somit seit langem bekannte Angriffsvektoren weiterhin ausgenützt werden können. Eine Publikation von Thomas et al. zeigt, dass durchschnittlich 87,7% aller Android Geräte anfällig sind für mindestens eine von elf bekannten und sicherheitskritischen Schwachstellen [12]. Des Weiteren stellt eine derartige Fragmentierungen Unternehmen vor große Herausforderungen, da sich die angebotene MDM-Funktionalität unterscheiden kann und sich oftmals große Unterschiede bei der Entwicklung mobiler Applikationen ergeben.

Google gibt monatlich die Liste der entdeckten Sicherheitsprobleme an Hersteller weiter. Einige Hersteller haben angekündigt diese Sicherheitsprobleme zeitnahe zu schließen, dazu gehört Google mit seinen Nexus Geräten [13], sowie Samsung [14], LG [15] und BlackBerry [16].

Version	*Codename*	*API*	*Anteil*
2.2	Froyo	8	0.1%
2.3.3 – 2.3.7	Gingerbread	10	2.7%
4.0.3 – 4.0.4	Ice Cream Sandwich	15	2.5%
4.1.x	Jelly Bean	16	8.8%

[42] Umgehen von Androids Face Unlock mithilfe von Fotos der Geräteinhaberin bzw. des Geräteinhabers: http://www.androidauthority.com/android-jelly-bean-face-unlock-blink-hacking-105556/ , letzter Zugriff am 11.04.2014. Aviv et al. zeigen des Weiteren wie anhand von Spuren am Display das Entsperrmuster erkannt werden kann [27].

[43] https://support.google.com/nexus/answer/6093922?hl=de

[44] https://play.google.com/store , letzter Zugriff am 14.04.2014

[45] http://www.welivesecurity.com/2015/09/22/android-trojan-drops-in-despite-googles-bouncer/

[46] Google, Aktuelle Verteilung der Plattform Versionen: http://developer.android.com/about/dashboards/index.html, letzter Zugriff am 08.03.2016

4.2.x		17	11.7%
4.3		18	3.4%
4.4	KitKat	19	35.5%
5.0	Lollipop	21	17.0%
5.1		22	17.1%
6.0	Marshmallow	23	1.2%

Tabelle 4: Android Verteilung der Plattform Versionen, Stand Februar 2016, Daten entnommen von: http://developer.android.com/about/dashboards/

4.4.6. Cloud-Anbindung

Systemeinstellungen, Kontaktdaten, WLAN-Passwörter etc. werden bei aktivierter Backupfunktionalität von Google auf zentralen Servern abgelegt. Die angebotene Backup-Funktionalität ist standardmäßig aktiviert, kann von der Benutzerin bzw. dem Benutzer jedoch jederzeit deaktiviert werden. Google bietet Applikationsentwicklerinnen und -entwicklern zusätzlich die Möglichkeit, über einen *Backup-Agenten*[47] Applikationsdaten auf zentralen Servern abzulegen. Der Backup-Mechanismus muss jedoch von der Entwicklerin bzw. vom Entwickler implementiert werden. Es ist anzumerken, dass die Ablage des Backups unverschlüsselt erfolgt und somit mitunter kritische Unternehmensdaten unverschlüsselt, je nach Gerät, auf Google Servern oder auf anderen Servern gespeichert werden[48].

Des Weiteren steht Entwicklerinnen und Entwicklern der *Google-Drive-Speicherbereich* der Benutzerin bzw. des Benutzers zur Verfügung. Bei beiden Optionen werden die Daten auf zentralen Servern von Google abgelegt. Zusätzlich zu den von Google angebotenen Varianten steht es Applikationen frei, Backups der Applikationsdaten auf eigenen Servern anzulegen.

4.4.7. MDM

Android verfügt seit der Version 2.2 über im Betriebssystem integrierte MDM-Funktionalität, erfordert jedoch die Installation eines zusätzlichen MDM-Agenten (einer herkömmlichen Applikation), um die vom Betriebssystem angebotenen Regeln konfigurieren zu können. Mehrere Drittanbieter bieten derartige MDM-Agenten für Android an[49]. Die angebotene Funktionalität ist stark abhängig vom jeweiligen Agenten, der die vom Betriebssystem zur Verfügung gestellte *Device Administration API*[50] anspricht. Aufgrund der Offenheit von Android steht es Geräteherstellern frei, zusätzliche MDM-Funktionalität zu integrieren. Geräte des Herstellers Samsung verfügen beispielsweise über eine größere Anzahl an MDM-Regeln als in Android standardmäßig vorgesehen[51]. Diese Fragmentierung führt zu Problemen beim Einsatz in Unternehmen, falls unterschiedliche Android Geräte verwaltet werden.

Die Basis-Android-Version unterstützt nur rudimentäre MDM-Regeln[52]:

- **Zugriffsschutz**
  - Passcode zum Entsperren des Geräts erforderlich
  - Einstellungen zur Komplexität, Länge und zeitlichen Gültigkeit des gewählten Passcodes
  - Aktivierung der Dateisystemverschlüsselung
- Unterbinden der Verwendung der **Kamera** am mobilen Gerät

[47] Dokumentation Google Backup Agent: http://developer.android.com/guide/topics/data/backup.html , letzter Zugriff am 19.05.2016

[48] Unverschlüsselte Ablage von WLAN-Passwörtern in der Google Cloud: http://www.heise.de/security/meldung/Android-und-die-Passwoerter-Offene-Tueren-fuer-Spionage-1917386.html , letzter Zugriff am 11.04.2014

[49] Airwatch MDM-Agent für Android: http://www.air-watch.com/solutions/android

[50] http://developer.android.com/guide/topics/admin/device-admin.html, letzter Zugriff am 11.04.2014

[51] Auflistung aller im Rahmen von Samsung SAFE zur Verfügung stehenden MDM-Möglichkeiten: http://www.samsung.com/us/business/samsung-for-enterprise/downloads/SAFE_Brochure_Updated_1012.pdf, letzter Zugriff am 11.04.2014

[52] Auflistung aller unterstützten MDM-Regeln: http://developer.android.com/guide/topics/admin/device-admin.html, letzter Zugriff am 19.05.2016

- **Remote Wipe**
- **Remote-Sperre** des Geräts

Da der MDM-Agent lediglich eine herkömmliche Android-Applikation darstellt, steht es Benutzerinnen und Benutzern frei, den MDM-Agenten jederzeit selbstständig zu deinstallieren und sich somit den MDM-Regeln zu entziehen. Des Weiteren obliegt es dem jeweiligen MDM-Agenten zu entscheiden, mit welchen Einschränkungen die Benutzerin bzw. der Benutzer zu rechnen hat, falls die geforderten MDM-Regeln nicht eingehalten werden. Bei der Installation eines MDM-Agenten wird die Benutzerin bzw. der Benutzer darüber in Kenntnis gesetzt, dass ein *Device Administrator* installiert wird und über welche Eingriffsmöglichkeiten dieser am Gerät verfügt. MDM-Agenten bieten oftmals zusätzliche Funktionalität wie die Lokalisierung des Geräts, Installation von Applikationen, die Verwaltung von Backups, etc.[53]

4.4.8. BYOD

Vor Android 5 verfügte Android über keine im Betriebssystem integrierte BYOD-Mechanismen. BYOD-Szenarien waren über die Verwendung eines MDM-Agenten möglich. Hierbei ergab sich jedoch ein großer Eingriff in die Privatsphäre der Benutzerin bzw. des Benutzers, da beispielsweise bei der Durchführung eines Remote-Wipes auch private Daten vom Gerät gelöscht werden. Seit Android 5 wird der BYOD-Fall durch *Android for Work* abgedeckt.

4.4.9. Android for Work

Mit Android 5 wurde das Konzept eines Gerätebesitzers und eines Profilbesitzers eingeführt. Dieses Konzept soll den Firmengeräte- und den BYOD-Anwendungsfall unterstützen. Das Konzept passiert auf dem mit Android 4.2 eingeführten Multiuser-Konzept. Das Multiuser-Konzept war für Fälle gedacht wo ein Gerät von zwei oder mehreren verschiedenen Benutzern verwendet wird. Jeder dieser Benutzer hat dabei seine eigenen Applikationen, UI und zum Teil auch eigenen Einstellungen. Das erste Konto *Primary User*, welches zum Gerät hinzugefügt wird hat besondere Rechte und kann nicht entfernt werden. Alle weiteren Benutzerkonten, *Secondary User,* können gelöscht werden, sowohl von der jeweiligen Besitzerin bzw. dem jeweiligen Besitzer als auch vom *Primary User*. *Secondary User* Konten sind voneinander getrennt und können sich nicht gegenseitig beeinflussen. Das *Primary User* Konto lauft ständig, auch wenn ein anderes Konto gerade im Vordergrund ist. *Secondary User* Konten laufen auch im Hintergrund weiter, können aber vom Betriebssystem gestoppt werden falls der Speicher knapp wird. Zudem können *Secondary User* Konten, die im Hintergrund laufen, keine UI zeigen und keine Bluetooth Dienste verwenden. Dieser Multiuser-Modus erlaubte es bereits private Apps von geschäftlichen Apps zu trennen. Dieses Konzept wurde mit *Android for Work* verbessert [17].

Android for Work führt bei verwalteten Geräten neben dem Gerätebesitzer und dem Profilbesitzer noch ein Arbeitsprofil[54] ein. Mittels des Arbeitsprofils können geschäftliche Daten und Apps von der Organisation verwaltet werden, während alle anderen Daten und Funktionen weiterhin unter der Kontrolle der Benutzer bzw. Benutzerinnen stehen. Arbeitsprofile erlauben die sichere Verwaltung von Arbeitsumgebungen ohne dass die Nutzer gehindert werden ihr Gerät wie gewünscht zu verwenden und eigene Apps und Daten abzulegen. Mit Hilfe von Arbeitsprofilen können Administratorinnen und Administratoren unter anderem folgende Aktionen ausführen:

- Kontopasswort ändern
- Unternehmenszugriff auf E-Mail-Server und Firmendaten verwalten
- Statistiken zum Arbeitsprofil einsehen
- Die Erstellung von Bildschirmaufnahmen im Arbeitsprofil verbieten
- Einschränken welche Daten zwischen dem privaten Profil und dem Arbeitsprofil geteilt werden dürfen

[53] Von Airwatch angebotene MDM-Funktionalität: http://www.air-watch.com/solutions/android, letzter Zugriff am 11.04.2014

[54] https://support.google.com/work/android/answer/6191949?hl=de

Die Geräteadministratorin bzw. der Geräteadministrator erhält auch einige Daten des verwendeten Geräts. Diese umfassen das verwendete Modell, die Seriennummer, die ID, die Telefonnummer, den Mobilfunkanbieter, das eingesetzte Betriebssystem, die Build-Nummer, die Kernel-Version, die Baseband-Version sowie die MAC-Adresse und die Sprache. Das Arbeitsprofil kann von der Device Policy Controller-App und dem Gerätebesitzer gelöscht werden. Alle lokalen Daten innerhalb des Profils werden dann gelöscht. Die privaten Dateien bleiben erhalten und können nur vom Gerätebesitzer gelöscht werden. Ebenso kann das Gerät nur vom Gerätebesitzer auf Werkseinstellungen zurückgesetzt werden. Ist das Unternehmen der Gerätebesitzer, handelt es sich um ein klassisches Firmentelefon. Ist das Gerät in privatem Besitz handelt es sich um den BYOD-Fall.

Android 5 bringt auch einige neue Policies [17], so kann nun beispielsweise festgelegt werden, dass immer eine VPN Verbindung vorhanden sein muss, oder dass der Traffic von bestimmten Benutzern oder für bestimmte Profile oder auch für bestimmte Applikationen über VPN geleitet wird. Mittels *Android for Work* kann nicht nur das Gerät verwaltet werden, es können auch Policies auf Applikationsebene gesetzt werden, sofern der Applikationsentwickler dies vorsieht.

Eine Liste der unterstützten Policies findet sich in [17]. Je nach Anwendungsfall werden nicht alle Policies unterstützt. Per Policy lassen sich beispielsweise die Passworteigenschaften, die Sperrzeiten und die Verwendung der Kamera regeln. Es können aber auch Zertifikate oder Schlüsselpaare installiert werden, Apps versteckt werden oder Benutzer entfernt werden.

4.5. Android – BlackBerry PRIV

Letzte Aktualisierung	Mai 2016

Mit PRIV will BlackBerry die Sicherheit für die BlackBerry steht zu Android bringen [16]. BlackBerry PRIV baute auf Android Lollipop (5.1) auf [18], seit Ende April kann auf Android 6.0 (Marshmallow) aktualisiert werden [19]. Android Lollipop kommt schon mit den im vorigen Abschnitt beschriebenen Sicherheitsfeatures. Android Marshmallow bringt zusätzlich noch ein überarbeitetes Berechtigungssystem. Wie bei iOS, kann der Zugriff auf Ressourcen nun zur Laufzeit freigeben oder blockiert werden. Zusätzlich hat BlackBerry einige weitere Sicherheitsfeatures hinzugefügt, die in den nächsten Abschnitten beschrieben werden.

Überblick – Änderungen im Vergleich zu Standard-Android	
Basissicherheit	Die Basissicherheitsfunktionen von Android wurden in mehreren Bereichen ergänzt: • Gehärteter Kernel • BlackBerry Integrity Detection System • Vertrauenswürdiger Herstellungsprozess • Hardware Vertrauensanker • BlackBerry Secure Compound • Enhanced Memory Protection • DTEK • Downgrade-Angriff wird verhindert
Verschlüsselung	FIPS 140-2 kompatibler Certicom/BlackBerry Kryptokernel: Android Dateisystemverschlüsselung, Schutz des Schlüsselmaterials in Blackberry Secure Compound (Trusted Execution Environment)
Zugriffsschutz	Zusätzlich werden folgenden Methoden zur Verfügung gestellt: • Picture Password • Keine Biometrie
Applikationsquellen	Keine Änderungen gegenüber der nicht modifizierten Android-Plattform
Updatesituation	BlackBerry stellt zeitnahe Updates bereit
MDM	Keine Ergänzungen gegenüber der nicht modifizierten Android Plattform

4.5.1. Basissicherheit

BlackBerry PRIV verwendet ein speziell gehärtes Android. Auf Hardware-Ebene setzt BlackBerry auf einen vertrauenswürdigen Herstellungsprozess, einen hardwarebasierten *Root of Trust* sowie auf BlackBerry Secure Compound (eine gehärtete Ausführungsumgebung). Auf Firmware-Ebene setzt BlackBerry auf einen Secure-Boot-Prozess sowie auf ein *Integrity Detection System*. Auf Android-Betriebssystemebene setzt BlackBerry auf *Kernel Hardening*, *Enhanced Memory Protectio*n und eine Dateisystemverschlüsselung. Auf App-Ebene stehen unter anderem DTEK, Picture Password, Password Keeper und BBM zur Verfügung (siehe weiter unten).

BlackBerry verwendet ein Ende-zu-Ende Herstellungsmodell um die Lieferkette, die Herstellungspartner und die Geräte zu verbinden. BlackBerry benutzt die hardware-basierten Schlüssel im Gerät um das Gerät zu verfolgen, zu verifizieren und zu provisionieren während es die einzelnen Herstellungsschritte durchläuft. Während dem Herstellungsprozess wird kryptografisches Material aufgebracht. Diese Schlüssel, die den Hardware-Vertrauensanker bilden, werden später für den Secure-Boot-Prozess und zur Geräteidentifizierung verwendet. PRIV Geräte verfügen über eine *Trusted Execution Enviroment*. Diese Umgebung heißt BlackBerry *Secure Compound* und schützt unter anderem den Secure-Boot-Prozess. Des Weiteren erlaubt BlackBerry *Secure Compound* die Ausführung von sicherheitskritischen Apps, wie der BlackBerry *Integrity Detection App* und ermöglicht die Generierung und Speicherung von vertraulichen Informationen wie Passwörtern und Schlüsseln. BlackBerry *Secure Compound* ist immun gegenüber Sicherheitsproblemen in Android,

da es auf einer tieferen Ebene als Android selbst läuft. Passwörter werden mittels PBKDF2 mit HMAC-SHA-512 abgeleitet. Informationen wie der Salt und die Anzahl der Iterationen werden im NVRAM abgelegt.

BlackBerry PRIV verwendet einen Secure-Boot-Prozess um sicherzustellen, dass nur von BlackBerry signierte Betriebssysteme geladen werden. Der Secure-Boot-Prozess besteht aus mehreren Stufen. Als Vertrauensanker dient ein Schlüssel, der bei der Herstellung in den Prozessor eingebettet wurde. In jeder Phase des Bootprozesses wird die nächste Stufe überprüft bevor sie ausgeführt wird. Schlägt die Überprüfung fehl, wird der Bootprozess abgebrochen. In der letzten Phase wird dann der Betriebssystemkern gestartet. Diese startet dann das Android-Betriebssystem. Der Secure-Boot-Prozess verhindert auch einen Downgrade-Angriff. Unter einem Downgrade-Angriff versteht man die Installation von älterer, aber gültiger Software, welche nicht alle Sicherheitsupdates enthält. Um die Sicherheit und Vertrauenswürdigkeit des Systems nach dem Bootvorgang zu gewährleisten verwendet BlackBerry ein Integrity Detection System. Das BlackBerry Integrity Detection System überwacht Events und Konfigurationsänderungen, welche auf eine Kompromittierung des Gerätes hindeuten könnten. Das BlackBerry Integrity Detection System überprüft auch die Mount-Rechte des Dateisystems, die Integrität des Kernels, die SELinux Policies und überwacht das Apps nicht erhöhte Rechte (z.B. durch „rooten“) erlangt haben. Enterprise Mobility Management (EMM) Lösungen können signierte Integritätsberichte anfordern. Der dazugehörige private Signaturschlüssel ist sicher im *Secure Compound* hinterlegt. Wird eine potentielle Kompromittierung eines Gerätes festgestellt, kann das betroffene Gerät beispielsweise zurückgesetzt werden, der Zugriff auf Arbeitsressourcen blockiert werden oder auch nur eine Warnung ausgegeben werden.

Android 6 bietet von Haus aus einige neue Sicherheitsfeatures wie eine USB Zugriffkontrolle oder ein modifiziertes Permission-System, wo zur Laufzeit Berechtigungen gegeben oder entzogen werden können. Zusätzlich hat BlackBerry den Linux Kernel gestärkt, indem beispielsweise nicht benötigter Code entfernt wurde sowie einige Kernel Sicherheitsupdates integriert wurden. Dadurch ergibt sich eine höhere Widerstandsfähigkeit gegen Malware. Unter *Enhanced Memory Protection* fasst BlackBerry die Änderungen am ASLR System zusammen. Android bietet bereits ASLR und eine Dateisystemverschlüsselung. BlackBerry verwendet und verbessert beide Systeme. ASLR soll das System gegen Exploits zu schützen, BlackBerry verwendet hierfür eigene Speicherlayout-Systeme. Die Dateisystemverschlüsselung wird im nächsten Abschnitt (4.5.2) behandelt.

Auf Applikationsebene bietet BlackBerry einige eigene Apps wie BBM, BlackBerry Password Keeper oder DTEK um die Sicherheit weiter zu erhöhen. Bei BBM handelt es sich um einen sicheren Sofortnachrichtendienst, welcher auch Sprach- und Videoanrufe unterstützt. Bei BlackBerry Password Keeper handelt es sich um eine Passwortmanager-App. Diese ermöglicht die sichere Speicherung von Passwörtern, Benutzernamen und Sicherheitsfragen bzw. deren Antworten. Die Daten werden mittels AES-256 verschlüsselt und sind durch ein Master-Passwort geschützt. Bei DTEK handelt es sich um eine Applikation, die die Sicherheit des Systems bewertet und Tipps zur Verbesserung gibt. Des Weiteren können Applikation auf bestimmte Verhaltensweisen beobachtet werden. Zu diesen Verhaltensweisen gehören die Benutzung der Kamera, der Zugriff auf Kontakte, die Abfrage des aktuellen Standorts, senden von Textnachrichten und die Aktivierung oder Deaktivierung des Mikrofons. Wird eine dieser Verhaltensweisen entdeckt, wird dies geloggt. Über die App können die Logeinträge pro App abgefragt werden.

4.5.2. Verschlüsselung

Seit Android 5 werden alle Benutzerdaten standardmäßig verschlüsselt abgelegt. Auch BlackBerry PRIV verwendet diesen Verschlüsselungsprozess, allerdings mit kleinen Anpassungen. Daten werden mit AES-128 (AES-CBC-ESSIV:SHA-256) verschlüsselt, der Verschlüsselungsschlüssel wird im BlackBerry *Secure Compound* abgelegt.

Zusätzlich zur Dateisystemverschlüsselung bietet PRIV einen Speicherkartenschutz an [20]. Je nach Konfiguration wird der Zugriff auf die Speicherkarte eingeschränkt oder komplett gesperrt. Bei nicht gemanagten Geräten kann nur das primäre Benutzerprofil auf die Speicherkarte zugreifen. Bei

Android for Work im Profilbesitzer Modus haben Applikationen aus dem Arbeitsprofil nur lesenden Zugriff auf die Speicherkarte. Applikationen außerhalb des Arbeitsprofils können sowohl lesend als auch schreibend auf die Karte zugreifen. Bei *Android for Work* im Gerätebesitzer-Modus gibt es nur einen Benutzer und dieser hat vollen Zugriff auf die Speicherkarte.

4.5.3. Zugriffsschutz

BlackBerry PRIV unterstützt im Wesentlichen die Standard Schutzmechanismen von Android. Das Gerät kann per PIN, Password, Muster oder Streichgeste gesperrt werden [21]. Auch Smart Lock wird unterstützt, wodurch das Gerät entsperrt bleibt, wenn das Gesicht oder die Stimme der Besitzerin bzw. des Besitzers erkannt wird. Weiteres kann dann das Gerät so eingerichtet werden, dass es an vertrauenswürdigen Orten oder in der Nähe von vertrauenswürdigen NFC- oder Bluetooth-Geräten entsperrt bleibt. Die Besonderheit von PRIV ist die Entsperrung per *Picture Password*. Dabei muss eine gewählte Zahl an eine gewählte Stelle von einem gewählten Bild bewegt werden. Auf einen Fingerabdruckscanner wurde bewusst verzichtet [22]. Wie auch bei anderen Android Geräten kann das Gerät aus der Ferne gesperrt werden, beispielsweise per Android Device Manager.

Per Enterprise Mobility Managment (EMM) können die Folgen von fehlerhaften Authentifizierungsversuchen definiert werden. Beispielsweise kann das Gerät konfiguriert werden, sich nach 10 falschen Passworteingaben zurückzusetzen. Dadurch werden alle Daten unwiederbringlich gelöscht. Im BYOD-Fall wird nur das Arbeitsprofil gelöscht.

4.5.4. Applikationsquellen

Als Applikationsquellen können der Google Play Store, der Google Play for Work Store sowie andere Drittanbieter Store verwendet werden [23] [24]. BlackBerry trennt allerdings vertrauenswürdige und nicht vertrauenswürdige Applikationen. Eine nicht vertrauenswürdige App kann beispielsweise nicht auf Daten von „Arbeits-Apps“ aus dem Google Play for Work Store zugreifen.

4.5.5. Updatesituation

Google erstellt jeden Monat eine Liste von entdeckten Sicherheitsschwachstellen (Security Bulletin) und übergibt diese Liste an BlackBerry und andere Android OEMs. Ungefähr ein Monat später wird die Liste veröffentlicht. BlackBerry hat angekündigt diese monatlichen Updates direkt an, über den BlackBerry Shop gekaufte, PRIV Geräte auszuliefern und an, am Programm teilnehmende, PRIV Wiederverkäufer [16]. Bei kritischen Sicherheitslücken kann BlackBerry sogenannte Hotfixes direkt an BlackBerry PRIV Geräte schicken. Kritische Hotfixes werden sofort ausgeliefert, es wird nicht auf das monatliche Updatefenster gewartet. Für Unternehmen bietet BlackBerry Enterprise-Managed Updates an. Dadurch kann das Unternehmen bestimmen wann und welche Updates installiert werden sollen. Im Vergleich zu anderen Android OEMs stellt BlackBerry Patches sehr zeitnah zur Verfügung. Für die Security Bulletins von Dezember 2015 bis März 2016 gelang es BlackBerry alle Sicherheitsschwachstellen zu schließen bevor sie von Google veröffentlicht wurden [25].

4.5.6. Cloud-Anbindung

BlackBerry PRIV Geräte können mit BES12 oder BES12 Cloud Servern verbunden werden. Bei BES12 handelt es sich um eine Enterprise Mobility Management Lösung, die im eigenen Firmennetz betrieben wird, während es sich bei BES12 Cloud [26] um eine Lösung handelt, welche von BlackBerry in deren Rechenzentren betrieben wird. Die Verbindung zum BES12 Server kann beispielsweise über BlackBerry Secure Connect Plus abgesichert werden.

4.5.7. MDM

PRIV unterstützt vier verschiedene Management-Optionen, die sowohl den BYOD und den COPE Fall abdecken.

Management-Option	Erklärung
MDM Kontrolle	Bei dieser Management-Option kann das Gerät nur über die Android IT-Administrationsbefehle und IT Regeln bzw. die von BlackBerry für PRIV bereitgestellten Funktionen gemanagt werden. Es wird kein separater Arbeitsplatz installiert, daher gibt es keine zusätzliche Sicherheit für Arbeitsdaten.
Container	Auf dem Gerät wird ein verschlüsselter Container mit eigenem Dateisystem aufgebracht. Arbeitsdaten und Arbeitsapps befinden sich innerhalb des Containers und werden somit geschützt. Der Zugang zum Container wird typischerweise per Passwort gesichert.
Android for Work: Profilbesitzer	Bei dieser Option, wird ein isoliertes Arbeitsprofil installiert. Eine Administratorin bzw. ein Administrator kann nur das Arbeitsprofil managen und dort Policies anwenden. Die Arbeitsdaten und Apps werden durch das Arbeitsprofil isoliert von privaten Daten und Apps.
Android for Work: Gerätebesitzer	Bei dieser Option hat das Gerät nur ein Profil und dieses wird von der Administratorin bzw. dem Administrator gemanagt. Dadurch kann die Administratorin bzw. der Administrator das gesamte Gerät managen und kontrollieren.

Die PRIV Geräte können mittels BES12 administriert werden. Unterstützt werden die Standard Android MDM Regeln und die *Android for Work* Regeln [27]. Mittels der MDM Regeln können die Passworteigenschaften vorgeben werden oder die Kamera deaktiviert werden. Über *Android for Work* stehen folgende Regeln zur Verfügung:

- Allow Bluetooth configuration
- Allow configuring mobile networks
- Allow tethering configuration
- Allow factory reset
- Allow mounting physical media
- Allow deleting users
- Allow SMS messages
- Allow USB file transfer
- Set time automatically
- Allow cross profile caller ID
- Allow installation of non Google Play apps
- Allow installing apps using debugging tools
- Require app verification
- Allow adding users
- Allow cross profile copy and paste
- Allow adding and removing accounts
- Allow screen capture

4.5.8. BYOD

Für den BYOD-Fall eignen sich besonders die Management-Optionen Container und *Android for Work* im Profilbesitzer Modus. Dadurch kann eine Administratorin bzw. ein Administrator die Arbeitsdaten und Apps entsprechend der Firmenregeln schützen bzw. managen, hat aber keinen Zugriff auf die privaten Dateien und Apps der Mitarbeiterin bzw. des Mitarbeiters.

4.6. Android – Samsung Knox

Samsung KNOX [28] baut auf Android auf und erweitert es um weitere Sicherheitsfeatures. Ziel von Samsung ist es Android unternehmenstauglich zu machen. Hierfür baut KNOX auf mehreren Schutzmechanismen auf, die zu unterschiedlichen Zeitpunkten ansetzen bzw. wirken. Google und Samsung haben 2014 eine Partnerschaft geschlossen, um Teile der KNOX Technologie in Android Lollipop zu integrieren [29].

Überblick – Änderungen im Vergleich zu Standard-Android	
Basissicherheit	Im Vergleich zum Standard-Android System werden weitere Basissicherheitsfunktionen eingefügt, die das System besser vor Manipulation durch Schadsoftware schützen. • Warranty Bit • Vertrauenswürdiger Herstellungsprozess • Hardware Vertrauensanker • Secure- & Trusted-Boot • KNOX Load-Time Defences • Dateibasierende dm-verity Implementierung • Periodic Kernel Measurement • Real-time Kernel Protection • Attestation • Downgrade-Angriff wird verhindert • TIMA KeyStore • TIMA CCM • Trusted UI • SSO Framework • Erweiterte SE for Android Version
Verschlüsselung	Samsung KNOX Systeme setzen auf hardwarebasierte Verschlüsselung. Ein wesentlicher Unterschied zu Standard-Android Geräten ist die Unterscheidung von unterschiedlichen Schutzklassen für das Verschlüsseln von Daten. • Hardwarebasierende Verschlüsselung • Protected & Sensitive Data
Zugriffsschutz	Im Vergleich zum Standard-Android System werden zusätzliche Authentifizierungsmaßnahmen integriert. Ein wesentlicher Punkt ist die mögliche Active Directory Anmeldung – vor allem für Geräte mit mehreren Benutzern (Shared Devices). • Unterstützung für Zwei-Faktor-Authentifizierung • KNOX Quick Access • Active Directory Anmeldung
Applikationsquellen	Installation aus unbekannten Quellen kann per MDM deaktiviert werden
Updatesituation	Einige Geräte erhalten monatliche Updates
MDM	Die MDM-Funktionalität ist ein wesentliches Unterscheidungsmerkmal zu Standard-Android. Es stehen im Vergleich dazu umfassende MDM-Regeln zur Verfügung die es ermöglichen die Geräte je nach Sicherheitsanforderung im Detail zu konfigurieren. • Über 1500 MDM APIs • Unterstützung für Shared Devices (Active Directory Anmeldung)

4.6.1. Basissicherheit

Samsung KNOX bietet viele Sicherheitsfeatures, die meisten davon beruhen auf der ARM TrustZone *Secure World*. Die *Secure World* ist ein isolierter Hardwarebereich, in dem sensitive Software ausgeführt wird. Normale Applikationen und der Kernel laufen in der *Normal World*. Die *Normal*

World hat keinen Zugriff auf die *Secure World*. Die *Secure World* hingegen kann auf Ressourcen aus der *Normal World* und aus der *Secure World* zugreifen. Samsung KNOX berücksichtigt Sicherheit in allen Phasen der Geräteentwicklung und Benutzung. Diese Phasen umfassen den Designzeitpunkt, den Herstellungszeitpunkt, den Boot-Zeitpunkt, den Softwareladezeitpunkt, die Laufzeit und den Update-Zeitpunkt.

Beispiele für Sicherheitsfeatures aus dem Designzeitpunkt sind beispielsweise die TrustZone und das *Warranty Bit*. Samsung KNOX Geräte enthalten ein *Warranty Bit*, dabei handelt es sich um eine Sicherung, die anzeigt ob auf dem Gerät jemals ein manipulierter Bootloader oder Kernel ausgeführt wurde. Sobald eine Manipulation festgestellt wird, wird diese Sicherung ausgelöst. Danach kann nicht mehr auf Samsung KNOX, die TrustZone *Secure World*, auf den *Device Root Key* und daher auf Arbeitsdaten zugegriffen werden.

Samsung erstellt und konfiguriert die Geräte in eigenen Fabriken. Daher hat Samsung die komplette Kontrolle über den Herstellungsprozess. Samsung nutzt den Herstellungsprozess um einen *Device Root Key (DRK)* und einen *Samsung Secure Boot Key (SSBK)* aufzubringen. Während der DRK einzigartig ist, teilen sich alle Samsung Geräte einen SSBK. Um sicherzustellen, dass nur vertrauenswürdige Software gestartet wird und um dies zu beweisen verwendet Samsung KNOX einen *Secure Boot* und einen *Trusted-Boot*-Prozess. Beim *Secure-Boot*-Prozess überprüft jede Komponente die Integrität der nächsten Komponente, bevor diese gestartet wird. Der *Trusted Boot* Prozess misst und speichert die Hash-Werte der nächsten Komponente. Die Werte werden in der TrustZone *Secure World* gespeichert und können später benutzt werden, um zu beweisen welche Versionen der Software gestartet wurden. Dadurch kann beispielsweise eine MDM-Lösung verifizieren, dass nur aktuelle sichere Softwareversionen geladen wurden. Dadurch können auch Downgrade-Attacken verhindert werden, bei denen z.B. eine ältere – aber gültig signierte – Softwareversion mit Sicherheitslücken installiert wird. Würde nur *Secure Boot* eingesetzt werden könnte so ein Angriff nicht verhindert werden, da die ältere Datei eine gültige Signatur enthält. Sowohl *Trusted Boot* als auch *Secure Boot* haben ihren Vertrauensanker in Hardware. Der primäre Bootloader wird von einem hardware-geschützen *Read-Only* Speicher geladen. Auch der SSBK, welcher zur Verifizierung der Signaturen benutzt wird, ist in einem nur einmal beschreibbaren Hardwarespeicher hinterlegt. Schlägt eine Signaturprüfung fehl, wird der Manipulationsversuch auf ewig festgehalten, indem die *Warranty Bit* Sicherung ausgelöst wird.

Die *KNOX Load-Time Defenses* beschützen die Softwareteile, die zur Bootzeit nicht verifiziert wurden. Wie auch Android benutzt KNOX dafür *dm-verity*. Während die Android *dm-verity* Implementierung blockbasierend arbeitet, funktioniert die Samsung Implementierung dateibasierend. Dadurch kann Samsungs Implementierung besser mit Firmware-Over-The-Air Updates umgehen. Mit *Secure Boot*, *Trusted Boot* und *dm-verity* wird sichergestellt, dass die Software zu dem Zeitpunkt, wo sie in den RAM Speicher geladen wird, authentisch und nicht kompromittiert ist. Zur Laufzeit verwendet *KNOX* die Sicherheitsfeatures *Periodic Kernel Measurement (PKM)*, *Real-time Kernel Protection (RKP) und Attestation*. PKM ist ein auf TrustZone basierendes Sicherheitsfeature, welches periodisch den Kernel auf unerwartete Veränderungen überprüft. RKP überwacht das Betriebssystem aus der TrustZone heraus in Echtzeit. Kritische Kernel Events werden abgefangen und inspiziert. Wird ein Ereignis entdeckt, welches einen unerlaubten Effekt auf den Kernel hat, wird dieses Event entweder gestoppt oder es wird protokolliert, dass es einen vermuteten Manipulationsversuch gab. RKP beschützt den Kernel vor bösartigen Modifikationen und vor Code-Injektion. Mittels *Attestation* kann ein Gerät gegenüber der MDM Lösung beweisen, dass es sich in einem vertrauenswürdigen Zustand befindet. Die *Attestation* Nachricht ist digital signiert und enthält unter anderem die Messwerte vom *Trusted Boot* Vorgang, den PKM Log, den RKP Log, den Zustand des *Warranty Bit*, den Modus von *Security Enhancement for Android* (SE for Android), Geräte Identifikationen sowie eine Einschätzung des Gerätes über den eigenen Zustand.

Um das Gerät auch bei Updates zu schützen, implementiert Samsung sogenannte Update-Time Defenses. Diese stellen sicher, dass beispielsweise keine zu alte Version installiert werden kann. Die minimale notwendige Softwareversion des Bootloaders ist in sicheren Hardwaresicherungen

hinterlegt (sogenannte Rollback Prevention Fuses). Die minimale notwendige Version des Kernels ist im Bootloader hinterlegt.

Neben diesen Features bietet Samsung KNOX noch weitere Sicherheitsfeatures wie einen *Trusted Boot-based KeyStore* (*TIMA KeyStore*), ein *Trusted Boot-based Client Certificate Managment (TIMA CCM)*, *SE for Android*, eine *Trusted UI*, sowie ein VPN und ein Single Sign-On (SSO) Framework. Auf den *TIMA KeyStore* sowie *TIMA CCM* wird im nächsten Abschnitt eingegangen. Samsung KNOX verwendet eine erweiterte Version von *Security Enhancement for Android* (SE for Android). Im Gegensatz zur Standard Version bietet Samsungs Version beispielsweise noch eine zwingend erforderliche Zugangskontrolle für APIs und eine *KNOX* Arbeitsbereich Isolierung für private Daten und Unternehmensdaten. Die *Trusted UI* erlaubt die Eingabe von sensitiven Daten wie PINs und Passwörtern. Die Daten, die über die *Trusted UI* eingegeben werden, sind geschützt vor dem Zugriff aus der *Normal World* und auch vorm Zugriff von nicht vertrauenswürdigen externen Geräten. Dazu erstellt die ARM Trustzone einen Hardwarepfad vom Bildschirm und dem Eingabegerät zur *Secure World*. Applikationen aus der *Normal World* können die Daten weder abfangen noch manipulieren. Um die Verbindung vom Gerät zum Unternehmensnetzwerk abzusichern, bietet KNOX unterschiedliche VPN Funktionen. Beispielsweise werden per App-VPN Verbindungen, Always-On oder On-Demand Verbindungen unterstützt. KNOX stellt auch ein SSO Framework bereit, dadurch kann ein Password benutzt werden um sich in den WorkSpace und in teilnehmende Apps einzuloggen. Samsung unterstützt dabei große Identitätsprovider wie Microsoft bzw. dessen Azure Active Directory.

Samsung KNOX bietet eine Trennung von privaten Apps und Workspace bezogenen Daten und Apps. Die Isolierung der Daten und Apps im Workspace von anderen Applikation stellt sicher, dass keine vertraulichen Informationen an nicht vertrauenswürdige Apps weitergereicht werden können. KNOX verbietet beispielsweise Kopieren und Einfügen, Interprozesskommunikation und andere Datenübertragungsmethoden zwischen Workspace Apps und normalen Apps.

4.6.2. Verschlüsselung

Wie auch Android unterstützt KNOX die Vollverschlüsselung des Datenträgers. Der Verschlüsselungsschlüssel ist ans Gerät gebunden. Zusätzlich unterstützt KNOX zwei Sicherheitsklassen für Daten die im KNOX Workspace erstellt werden, geschützte Daten (*Protected Data*) und sensitive Daten (*Sensitive Data*). Alle Daten, die im Workspace generiert werden, werden automatisch geschützt. Geschützte Daten werden immer verschlüsselt abgelegt und vor Zugriffen außerhalb des Workspaces geschützt. Der Verschlüsselungsschlüssel wird mit dem *Device-Unique Hardware Key* (DUHK) verschlüsselt. Daher kann der Verschlüsselungsschlüssel nur auf demselben Gerät entschlüsselt werden. Sensitive Daten werden auch immer verschlüsselt gespeichert. Zusätzlich bleiben sensitive Daten verschlüsselt solange der Workspace gesperrt ist. Der Verschlüsselungsschlüssel für sensitive Daten (*Container Master Key*, kurz CMK) wird ebenfalls mit dem DUHK verschlüsselt, zusätzlich gibt es aber eine Abhängigkeit zum PIN oder Muster des Workspaces. Für den Fall, dass der PIN bzw. das Muster vergessen wird kann der CMK auch mittels MDM entsperrt werden. Sensitive Daten die empfangen werden, während der Workspace gesperrt ist (z.B.: e-Mails), werden temporär mittels eines asymmetrischen Public-Key Verfahrens verschlüsselt. Wird der Workspace entsperrt, werden die zuvor empfangen Daten entschlüsselt und dann anschließend symmetrisch, mittels des Schlüssels für sensitive Daten, verschlüsselt. Daten die in der *KNOX Chamber* abgelegt werden, einem speziellen Verzeichnis im Dateisystem, werden automatisch als sensitive Daten behandelt. Weiteres bietet KNOX einen *Trusted Boot based KeyStore* (TIMA Keystore) sowie ein *Trusted Boot based Client Certificate Managment* (TIMA CCM). Der *TIMA KeyStore* stellt dieselben APIs bereit wie der *Android KeyStore*. Im Gegensatz zum *Android KeyStore* steht der *TIMA KeyStore* nur zur Verfügung, wenn der *Trusted Boot* Prozess bekannte gute Werte liefert und das *KNOX Warranty Bit* intakt ist. Dadurch können kryptografische Operationen nur ausgeführt werden, wenn sich das System in einem guten, vertrauenswürdigen Zustand befindet. Schlüssel im *TIMA KeyStore* werden mit dem *DUHK* verschlüsselt und können nur in der *TrustZone Secure World* entschlüsselt werden. Der *TIMA KeyStore* wird standardmäßig für *Android for Work Managed Profiles* benutzt. Alle kryptografischen Operationen werden in der TrustZone Secure World ausgeführt. *TIMA CCM* ermöglicht die Speicherung und Benutzung von

Zertifikaten. *TIMA CCM* kann mit einer SmartCard verglichen werden und unterstützt unter anderem die Verschlüsselung, Entschlüsselung, Signierung und Verifizierung von Daten. *TIMA CCM* kann über den PKCS#11 Standard angesprochen werden und steht ebenso wie der TIMA KeyStore nur zur Verfügung, wenn sich das System in einem guten, vertrauenswürdigen Zustand befindet.

4.6.3. Zugriffsschutz

Samsung KNOX bietet die von Android bekannten Sperrbildschirme. Zusätzlich kann der Zugriff auf den KNOX Workspace per PIN, Passwort oder Biometrie gesichert werden. KNOX unterstützt auch eine Zwei-Faktor-Authentifizierung. Beispielsweise kann der Fingerabdruck als erster Faktor benutzt werden und ein PIN oder Passwort als zweiter Faktor. Neben Googles SMART Lock, welches zum schnellen entsperren des Sperrbildschirms dient, unterstützt Samsung auch KNOX Quick Access. KNOX Quick Access ermöglicht den schnellen Zugriff auf den KNOX Workspace. KNOX Quick Access funktioniert aktuell mit Samsung Galaxy S6 Geräten. Befindet sich ein registriertes Samsung Gear Gerät in der Nähe, wird die Entsperrdauer des Workspaces verlängert. Dadurch muss sich die Benutzerin bzw. der Benutzer nicht so oft authentifizieren.

4.6.4. Applikationsquellen

Applikationen können über Google Play, Google Play for Work oder über den Samsung App Store installiert werden. Zusätzlich können mittels MDM Lösung private Unternehmensapplikationen installiert werden [30]. Die Installation von Applikationen aus unbekannter Quelle kann per Regel verboten werden [31].

4.6.5. Updatesituation

Im August 2015 hat Samsung angekündigt, ähnlich wie Google, einen Android-Sicherheits-Update-Prozess aufzusetzen. Diese Updates sollten regelmäßig, ungefähr einmal im Monat erscheinen [14]. Im Oktober hat Samsung einen Mobile Security Blog gestartet [32]. Über den Blog kann man Sicherheitsprobleme melden oder sich über aktuelle Bedrohungen informieren. Zudem führt der Blog die Samsung Geräte an, die ein monatliches Update bekommen [33].

4.6.6. Cloud-Anbindung

Samsung KNOX Geräte können mittels der cloudbasierten Verwaltungslösung Samsung SDS CellWe EMM verwaltet werden [34]. Samsung KNOX Geräte können auch ohne Cloud-Verbindung betrieben werden, dazu muss der KNOX Lizenz Server im eigenen Netzwerk aufgesetzt werden. Samsung bezeichnet dieses Feature als *No Mandated Cloud Connection* [35].

4.6.7. MDM

Samsung KNOX bietet über 1500 MDM APIs sowie eine Active Directory Integration. Folgende Kategorien von MDM APIs stehen zur Verfügung:

- Enterprise IT Compatibility
    - Account Management using blacklisting/whitelisting
    - Active Directory integration
    - LDAP Management
    - Enterprise Billing
    - VPN
- Security and Compliance
    - Device Admin Management
    - Firewall
    - Password Management
    - Device Security
    - Remote Event Injection
    - Audit Logging
    - Usability

    - Kiosk Mode
    - Workspace Management
    - Multiuser Mode
- Device Control
    - Date and Time
    - Bluetooth
    - Location Management
    - Device Restrictions
    - WiFi Configurations
    - APN Settings
    - Device Inventory
- Application Management
    - Browser
    - Email/Exchange Configuration
    - Application Management
- Telephony
    - Telephony Management
    - SIM Change Information
    - Roaming Restrictions

Speziell für Unternehmensgeräte unterstützt Samsung sogenannte Shared Devices. Wird ein Gerät zu einem Shared Device konfiguriert, kann es von mehreren Angestellten verwendet werden. Die Angestellten können Sich beispielsweise über Ihre *Active Directory ID* anmelden. Aus Gründen der Sicherheit und der Privatsphäre werden alle Benutzerdaten gelöscht sobald sich die Benutzerin bzw. der Benutzer rausloggt.

4.6.8. BYOD

Samsung KNOX unterstützt sowohl den Corporate-Owned Personally Enabled (COPE) als auch den Bring Your Own Device (BYOD) Anwendungsfall. Speziell für diese Fälle, unterstützt Samsung KNOX *Enterprise Billing*. *Enterprise Billing* ist ein Mechanismus, der den privaten Datenverbrauch vom geschäftlichen Datenverbrauch trennt. Dadurch kann das Unternehmen die Besitzerin bzw. den Besitzer des Gerätes im BYOD Fall für den angefallenen Datenverbrauch entschädigen. Im COPE Fall ermöglich *Enterprise Billing*, dass das Unternehmen die Kosten für den privaten Datenverbrauch an die Angestellte bzw. den Angestellten weiter verrechnen kann.

5. Literaturverzeichnis

[1] Apple, „iOS Security," 09 2015. [Online]. Available: http://www.apple.com/business/docs/iOS_Security_Guide.pdf. [Zugriff am 10 03 2016].

[2] Apple, „Deploying iPhone and iPad Mobile Device Management," 2012. [Online]. Available: https://www.apple.com/nz/ipad/business/docs/iOS_6_MDM_Sep12.pdf. [Zugriff am 10 03 2016].

[3] A. Meeus, „Windows 10 Mobile security guide," 19 05 2016. [Online]. Available: https://technet.microsoft.com/en-us/itpro/windows/keep-secure/windows-10-mobile-security-guide. [Zugriff am 19 05 2016].

[4] Microsoft Corp., „Assurance Activity - Microsoft Windows 10 Mobile with Lumia 950, 950 XL, 550, 635, and Windows 10 with Surface Pro 4," 29 04 2016. [Online]. Available: https://www.niap-ccevs.org/st/st_vid10694-aar.pdf. [Zugriff am 23 05 2016].

[5] Microsoft Corp., „Security Target - Microsoft Windows 10 Mobile with Lumia 950, 950 XL, 550, 635, and Windows 10 with Surface Pro 4," 12 04 2016. [Online]. Available: https://www.niap-ccevs.org/st/st_vid10694-st.pdf. [Zugriff am 23 05 2016].

[6] BlackBerry, „BlackBerry Enterprise Service 10. BlackBerry Device Service Solution Version 6.2," [Online]. Available: http://docs.blackberry.com/en/admin/deliverables/49294/BlackBerry_Device_Service_6.2_Security_Technical_Overview_en.pdf. [Zugriff am 2013].

[7] Pulser_G2, „A Look at Marshmallow Root & Verity Complications," 07 10 2015. [Online]. Available: http://www.xda-developers.com/a-look-at-marshmallow-root-verity-complications/. [Zugriff am 18 05 2016].

[8] Android, „Security Enhancements in Android 5.0," [Online]. Available: https://source.android.com/security/enhancements/enhancements50.html. [Zugriff am 19 05 2016].

[9] Android, „Security Enhancements in Android 6.0," [Online]. Available: https://source.android.com/security/enhancements/enhancements60.html. [Zugriff am 19 05 2016].

[10] Android, „Full Disk Encryption," [Online]. Available: https://source.android.com/security/encryption/. [Zugriff am 18 05 2016].

[11] P. Teufl, A. Fitzek, D. Hein, A. Marsalek, A. Oprisnik und T. Zefferer, „Android Encryption Systems," International Conference on Privacy & Security in Mobile Systems, 2014.

[12] D. R. Thomas, A. R. Beresford und A. Rice, „Security Metrics for the Android Ecosystem," [Online]. Available: https://www.cl.cam.ac.uk/~drt24/papers/spsm-scoring.pdf. [Zugriff am 19 05 2016].

[13] Android, „Official Android Blog: An Update to Nexus Devices," 05 08 2015. [Online]. Available: http://officialandroid.blogspot.co.at/2015/08/an-update-to-nexus-devices.html. [Zugriff am 19 05 2016].

[14] Samsung Electronics Co,.Ltd, „Samsung Announces an Android Security Update Process to Ensure Timely Protection from Security Vulnerabilities," 05 08 2015. [Online]. Available: https://news.samsung.com/global/samsung-announces-an-android-security-update-process-to-ensure-timely-protection-from-security-vulnerabilities. [Zugriff am 27 01 2016].

[15] E. Dreyfuss, „Big Android Makers Will Now Push Monthly Security Updates | WIRED," 06 08 2015. [Online]. Available: https://www.wired.com/2015/08/google-samsung-lg-roll-regular-android-security-updates/. [Zugriff am 19 05 2016].

[16] D. Kleidermacher, „Managing Android Security Patching for PRIV," 04 11 2015. [Online]. Available: http://blogs.blackberry.com/2015/11/managing-android-security-patching-for-priv/. [Zugriff am 05 04 2016].

[17] Google, „Android for Work - Security," 05 2015. [Online]. Available: https://www.google.com/work/android/files/android-for-work-security-white-paper.pdf. [Zugriff am 19 05 2016].

[18] D. Kleidermacher, „Why BlackBerry's Android is Best for Security and Privacy," 24 11 2015. [Online]. Available: http://blogs.blackberry.com/2015/11/why-blackberrys-android-is-best-for-security-and-privacy/. [Zugriff am 29 04 2016].

[19] V. C. Walker, „Sweet! Android 6.0 Marshmallow Now Available on PRIV," 26 04 2016. [Online]. Available: http://blogs.blackberry.com/2016/04/sweet-android-6-0-marshmallow-now-available-on-priv/. [Zugriff am 17 05 2016].

[20] BlackBerry, „Media card protection - Sicherheitshandbuch für BlackBerry powered by Android - latest," 25 04 2016. [Online]. Available: https://help.blackberry.com/de/security-guide-for-blackberry-powered-by-android/latest/security-guide-for-blackberry-powered-by-android-html/kja1437675778481.html. [Zugriff am 17 05 2016].

[21] BlackBerry, „Sperren, Kennwörter und Schützen Ihrer Daten - PRIV - 6.0," [Online]. Available: http://help.blackberry.com/de/priv/current/help/security-settings.html. [Zugriff am 29 04 2015].

[22] E. Protalinski, „BlackBerry talks Priv security, privacy, and why Android now," 6 11 2015. [Online]. Available: http://venturebeat.com/2015/11/06/blackberry-talks-priv-security-privacy-and-why-android-now/ . [Zugriff am 29 04 2015].

[23] lgazzola, „Google Play Apps Now Available Via PRIV," 11 06 2016. [Online]. Available: http://blogs.blackberry.com/2015/11/google-play-apps-now-available-via-priv/. [Zugriff am 29 04 2016].

[24] R. Brandom, „What the BlackBerry Priv means for Android security," 06 11 2015. [Online]. Available: http://www.theverge.com/2015/11/6/9680698/blackberry-priv-patch-android-security. [Zugriff am 17 05 2016].

[25] D. Kleidermacher, „Beating Expectations: Android Security Patching for PRIV," 11 03 2016. [Online]. Available: http://blogs.blackberry.com/2016/03/beating-expectations-android-security-patching-for-priv/. [Zugriff am 05 04 2016].

[26] BlackBerry, „Sicherheitsmerkmale von BES12 Cloud," [Online]. Available: http://help.blackberry.com/de/bes12-cloud/latest/security/awi1420835482948.html. [Zugriff am 17 05 2016].

[27] BlackBerry, „Policy Reference Spreadsheet BES12 12.4.1," [Online]. Available: http://help.blackberry.com/en/bes12/current/policy-reference-spreadsheet-zip/Policy_Reference_Spreadsheet_BES12_12.4_en.zip. [Zugriff am 23 05 2016].

[28] SAMSUNG, „Whitepaper: Samsung KNOX Security Solution," 03 2016. [Online]. Available: https://www.samsungknox.com/en/system/files/whitepaper/files/Samsung_KNOX_Security_Solution_V1_10_0.pdf. [Zugriff am 19 04 2016].

[29] SAMSUNG, „Samsung and Google to Bring Enterprise Enhancements to Android," 26 06 2014. [Online]. Available: http://www.samsungmobilepress.com/2014/06/26/Samsung-and-Google-to-Bring-Enterprise-Enhancements-to-Android-1. [Zugriff am 29 04 2016].

[30] SAMSUNG BUSINESS, „In-Depth Look at Capabilities:," [Online]. Available: http://www.samsung.com/hu/business-images/insights/2015/Samsung_KNOX_and_Android_for_Work_2-0-0.pdf. [Zugriff am 27 04 2016].

[31] SAMSUNG, „What does the "Permit installation of non-Google Play apps" setting in CellWe EMM do?," [Online]. Available: https://www.samsungknox.com/en/faq/what-does-permit-installation-non-google-play-apps-setting-cellwe-emm-do. [Zugriff am 27 04 2016].

[32] SAMSUNG, „Welcome to Samsung Mobile Security Blog," [Online]. Available: http://security.samsungmobile.com/. [Zugriff am 27 04 2016].

[33] SAMSUNG, „Introduction to Samsung Android Security Updates," [Online]. Available: http://security.samsungmobile.com/introsm.html. [Zugriff am 27 04 2016].

[34] SAMSUNG, „Eine einzige Verwaltungskonsole mit CellWe EMM," [Online]. Available: https://www.samsungknox.com/de/products/knox-emm. [Zugriff am 27 04 2016].

[35] SAMSUNG, „Android security maximized by Samsung KNOX," [Online]. Available: http://www.samsung.com/ie/business-images/insights/2015/Android_securityKNOX_online-0.pdf. [Zugriff am 27 04 2016].

[36] C. Orthacker, P. Teufl, S. Kraxberger, G. Lackner, M. Gissing, A. Marsalek, L. Johannes und O. Prevenhuber, „Android Security Permissions – Can we trust them?," in *Security and Privacy in Mobile Information and Communication Systems: Third International ICST Conference, MobiSec 2011, Aalborg, Denmark, May 17-19, 2011, Revised Selected Papers*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2012, pp. 40-51.

[37] P. Teufl, T. Zefferer und C. Stromberger, „Mobile Device Encryption Systems," in *28th IFIP TC-11 SEC 2013 International Information Security and Privacy Conference*, 2013.

[38] P. Teufl, T. Zefferer, C. Stromberger und H. Christoph, „iOS Encryption Systems - Deploying iOS Devices in Security-Critical Environments," in *SECRYPT*, 2013.

[39] A. J. Aviv, K. Gibson, E. Mossop, B. Matt und J. M. Smith, „Smudge Attacks on Smartphone Touch Screens," in *USENIX conference on Offensive technologies*, 2010.