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

ANALYSE VON FAHRZEUG-APPS

STUDIE, VERSION VOM 18.9.2016

Dominik Ziegler – dominik.ziegler@a-sit.at

Zusammenfassung: Die anhaltende Popularität mobiler Kommunikationstechnologien führt dazu, dass diese seit einigen Jahren verstärkt auch im Automotive-Bereich zum Einsatz kommen. Ein Beispiel hierfür sind mobile Apps, die es Lenkerinnen und Lenkern erlauben, mit ihrem Auto zu interagieren. Die Ver- und Entriegelung des Fahrzeugs oder das Remote-Starten von Standheizungen und Klimaanlagen sind nur einige der Use Cases, die von diesen Applikationen unterstützt werden. Mehrere Vorfälle in der jüngeren Vergangenheit zeigten jedoch auf, dass die von Automobilherstellern zur Verfügung gestellten Apps mitunter Angriffen nicht standhalten und so die Sicherheit des Gesamtsystems gefährden. Damit stellen mobile Anwendungen im Automotive-Bereich ein neues potentielles Anwendungsgebiet dar, das durch den richtigen Einsatz sicherer Informations- und Kommunikationstechnologien profitieren kann.
Diese Studie erhebt den aktuellen Stand der Technik und Forschung mobiler Anwendungen für Fahrzeuge. Sie gibt einen Überblick über populäre mobile Begleitapplikationen und deren Funktionsumfang. Weiters behandelt sie die verwendeten Technologien und Authentifizierungsmechanismen. Die so erhobenen Daten werden benutzt, um mögliche Risiken und Angriffsvektoren mobiler Anwendungen zu identifizieren. Ergänzt wird diese Studie durch die gezielte Analyse verfügbarer Apps von Automobilherstellern. Die im Rahmen dieses Projekts erzielten Resultate zeigen, dass mobile Anwendungen für Autos durchaus anfällig für gezielte Angriffe sein können und Verbesserungspotential besteht. Im Sinne Responsive Disclosure werden zwar mögliche Schwachstellen skizziert, aber nicht ob und bei welche App diese gefunden wurden. Dies wird Herstellern mitgeteilt.

Inhaltsverzeichnis

1. Einleitung

Mit der Omnipräsenz von Smartphones hat sich in den letzten Jahren ein Trend in Richtung Mobilität und Konnektivität entwickelt. Die Nachfrage nach Entertainment bzw. Zugang zum Internet macht auch vor dem Automotive-Sektor nicht halt. Die meisten großen Auto-Hersteller sind deswegen bereits auf diesen Zug aufgesprungen und erweitern die Funktionalität bzw. Möglichkeiten ihrer Fahrzeuge mit Hilfe von Smartphones. Was folgt sind Autos, die längst nicht mehr nur reine Fortbewegungsmittel sind.

Praktisch jeder große Hersteller nutzt deswegen die Präsenz von Smartphones und bietet mittlerweile mobile Applikationen an, die es erlauben, mit dem Fahrzeug zu interagieren. Diese mobilen Begleitapplikationen dienen primär als Erweiterung zu bestehenden System, Hilfestellung zum Fahrzeug oder als zusätzlicher Informationsfaktor. So erlauben sie es beispielsweise, Zustände von Fahrzeugen, wie den Ladezustand der Batterie, die aktuelle Tankfüllung oder den Kilometerstand auszulesen sowie das Vehikel zu kontrollieren, dieses aus der Ferne über das Internet zu entriegeln, den Motor zu starten oder die Klimaanlage zu steuern. Ein weiterer Anwendungsfall sind außerdem Car-Sharing oder Autovermietungen. Eine App kann dabei das Smartphone zum passenden Schlüssel machen.

Jüngste Vorfälle zeigen jedoch, dass die von Automobilherstellern zur Verfügung gestellten Apps mitunter Angriffen nicht standhalten und so die Sicherheit des Gesamtsystems gefährden [1]. Durch die zur Verfügung gestellte Funktionalität stellen mobile Anwendungen im Automotive-Sektor also nicht nur Gewinn dar, sondern können zu undefiniertem Verhalten am Fahrzeug bis hin zur ungewollten Kontrolle von außen oder Dritten führen. Aus diesem Grund erhebt die vorliegende Studie die aktuelle Situation und gibt einen Überblick über den aktuellen Stand der Technik.

Die Studie gliedert sich in 6 Abschnitte. Sie behandelt dabei in Abschnitt 2 die Grundlagen von mobilen Anwendungen, wie deren Steuerung oder Kommunikation. Dies soll ein Grundverständnis über diskutierte Technologien und Konzepte schaffen und dient als Basis für die weiteren Kapitel. Abschnitt 3 gibt im Anschluss detaillierte Information über existierende Anwendungen und deren Funktionsumfang. Dabei werden die verwendeten Technologien behandelt, sowie der Stand der Technik und die verwendeten Authentifizierungsmechanismen analysiert. Außerdem wird der Funktionsumfang von populären Auto-Apps aufgelistet. Dies erlaubt in weiterer Folge eine einfache Kategorisierung in mögliche Angriffsszenarien. Im Anschluss werden allgemeine Risiken, die mobile Auto-Apps mit sich bringen, diskutiert. Dies beinhaltet mögliche Angriffsvektoren, deren Auswirkungen sowie eine Diskussion über deren Umsetzbarkeit. Im letzten Abschnitt wird eine Analyse von populären Remote-Apps durchgeführt. Dabei werden die verwendeten Authentifizierungsmechanismen bzw. eventuelle Schwachstellen diskutiert.

2. Grundlagen

Ein „Smart-Vehikel“ bezeichnet im Allgemeinen ein Fahrzeug, welches durch Kombination von verschiedenen Kommunikationstechnologien, Computern oder Daten Benutzerinnen und Benutzern eine Vielzahl an Informationen oder Funktionalität zur Verfügung stellt. Ziel ist es, den Komfort, die Sicherheit oder die Benutzerfreundlichkeit zu steigern. Diese zusätzlichen Funktionen lassen sich in mehrere Anwendungsfälle kategorisieren [2]:

- **Kommunikation**: Mithilfe von Netzwerk-Funktionalität via mobilen Netzwerken können der Benutzerin und dem Benutzer verschiedene Services wie z.B. Internet, SMS, Notfall Services etc. zur Verfügung gestellt werden.

- **Navigation**: Systeme wie GPS werden für Navigation, die Feststellung des aktuellen Vehikel-Standortes, das Anzeigen von Parkplätzen in der Nähe oder der Routenplanung verwendet.

- **Assistenz-Systeme**: Zusätzliche Informationen (Ölstand, Reifendruck, etc.) über das

Fahrzeug können abgerufen werden. Dies beinhaltet auch Verkehrsinformationen oder fällige Service-Termine.

- **Infotainment**: Ziel eines Infotainments-Systems ist es, Informationen wie Verkehr, Wetter, etc. zur Verfügung zu stellen bzw. die Unterhaltungsmöglichkeiten (Musik, Streaming usw.) zu steuern

- **Sicherheit:** Smart-Vehikel können durch verschiedene Maßnahmen z.B. Melden von Fehlfunktionen, Unfällen oder Straßensperren und Inter-Vehikel-Kommunikation die Sicherheit für die Insassen oder Dritte erhöhen.

In Kombination mit Smartphones, respektive mobilen Anwendungen, ergeben sich durch diese Anwendungsfälle neue Möglichkeiten. Dabei werden Apps verwendet, um einerseits Informationen über das Fahrzeug abzurufen, andererseits Kommandos an dieses zu schicken. Praktisch jeder große Hersteller bietet bereits Apps an, mit deren Hilfe mit dem Fahrzeug interagiert werden kann. Ebenso gibt es „Nachrüst-Kits“ von Drittherstellern für ältere Fahrzeuge. Diese interagieren über die OBD[1] Schnittstelle direkt mit dem Fahrzeug.

Im Grunde gibt es je nach Applikation und Hersteller verschiedene Verbindungstypen (Bluetooth/USB, WLAN, Internet) und in Folge dessen unterschiedliche verfügbare Optionen. Als Beispiel sei hier die Remote Steuerung von Fahrzeugen genannt. Dabei werden Befehle an das Fahrzeug (meist über das Internet) gesendet, ohne dass sich die Benutzerin und der Benutzer in dessen unmittelbarer Nähe befinden muss. Typische Kommandos dafür sind beispielsweise das Entriegeln der Türen oder die Fernsteuerung der Klimaeinheit. Außerdem setzen Hersteller auf USB oder Bluetooth-Verbindungen, um Funktionen zu nutzen, die meist nur unmittelbar vor oder während dem Fahrbetrieb sinnvoll erscheinen, und eine Präsenz der Insassen voraussetzen. Eine mögliche Steuerung der Musikeinheit findet deswegen meist über diese „In-Vehicle-Connectivity“ Verbindungen statt. Ebenso können manche Fahrzeuge via Bluetooth entsperrt werden.

Diese zusätzliche Funktionalität verschafft zwar neue Möglichkeiten, ermöglicht aber auch neue Angriffsvektoren. Während bei Verbindungen über Bluetooth, USB oder WLAN meist eine (unmittelbare) physische Präsenz des Angreifers zum Fahrzeug vorausgesetzt wird, ermöglicht eine Verbindung über das Internet weitreichende Angriffe, da sich der Angreifer nicht mehr in direkter Nähe zum Vehikel befinden muss. Komponenten, die ursprünglich isoliert und nicht für die Steuerung über das Internet konzipiert und entwickelt worden waren, müssen dementsprechend abgesichert werden. Dies fordert einen sichereren Einsatz von Informations- und Kommunikationstechnologien und ein Überdenken der existierende Sicherheits-Anforderungen, um Schäden am Fahrzeug und Person zu verhindern.

3. Anwendungen

Durch die Kombination von Smart-Vehicles mit Smartphones ergeben sich verschiedene Funktionen, die mit Hilfe von mobilen Begleitapplikationen genutzt werden können. Diese kann man im Grunde in zwei Kategorien einteilen: Non-Intrusive (z.B. Abrufen von Status-Informationen) und Intrusive (z.B. Manipulation am Fahrzeug, Motor starten) Funktionen. Je nach Hersteller, Fahrzeug und Ausstattung ergeben sich dabei unterschiedliche Anwendungen und Möglichkeiten. Während Intrusive-Funktionen in den meisten Fällen nur in Kombination mit dem Internet verfügbar sind und deswegen auch aus der Ferne genutzt werden können, können Non-Intrusive Funktionen meist unabhängig vom Verbindungstyp genutzt werden.

Für die Steuerung von Fahrzeugen bzw. das Auslesen von Informationen über das Fahrzeug genügt in einer Vielzahl der Fälle eine direkte (USB, Bluetooth, WLAN) Verbindung zum Fahrzeug, obgleich dieselbe Funktion ebenso über eine indirekte Verbindung (Kommunikation von Smartphone und Auto via einen Server) erreicht werden kann. Da die meisten modernen Smartphones ohnehin diese Schnittstellen besitzen, kommen andere Technologien wie NFC bis jetzt kaum zum Einsatz.

[1] On-Board-Diagnose

Dieses Kapitel gibt einen Überblick über populärer Begleitapplikationen für Android. Bei den genannten Applikationen gibt es ebenso Versionen für iOS, weswegen der Funktionsumfang dieser Apps zusammengefasst wurde. Ebenso werden die genutzten Authentifizierungsmethoden im Detail diskutiert.

3.1. Funktionen

Im Allgemeinen beschränken sich mobile Begleitapplikation auf die Kommunikation mit dem Fahrzeug und dienen primär der Informationsbeschaffung oder der Steuerung des Autos. In Einzelfällen können damit aber auch externe Geräte (z.B. Gargentüröffner) angesprochen und konfiguriert werden. Da diese Funktionen allerdings nicht direkt mit dem Fahrzeug in Verbindung stehen, zählen sie nicht zum Umfang dieser Studie. Abbildung 1 gibt einen Überblick über populäre Anwendungen. Deren Funktionsumfang wurde in Non-Intrusive sowie Intrusive Funktionen unterteilt sowie die verwendete Kommunikationstechnologie gesondert notiert. Durch diese Unterteilung lässt sich in weiterer Folge eine Zuweisung in verschiedene Risikogruppen (siehe Kapitel 4) erreichen.

Feature	com.acura.acuralink.connect	com.honda.hondalink.connect	de.audi.immiapp	com.psa.citroen	com.bmwi.remote	com.bmw.remote	com.daimler.mm.android	com.daimler.mbfa.android	com.directed.android.viper	com.fiat.bev.access	com.ford.nafordowner	com.ford.remoteaccess	com.gm.chevrolet.nomad.ownership	com.gm.cadillac.nomad.ownership	com.gm.chevrolet.nomad.ownership	com.gm.buick.nomad.ownership	com.stationdm.bluelink	com.landrover.incontrolremote	com.us.lexusenformremote	com.mazda.mymazdaapp	com.digitas.android.nissan.carwings	com.nissan.infiniticonnection	cz.skodaauto.mfapro	com.bosch.rcconn	com.teslamotors.tesla	de.volkswagen.carnet.eu.eremote	de.volkswagen.mediacontrol	de.volkswagen.appconnect.rulez
Status (Non-Intrusive)																												
Benzin Stand	x			x			x	x										x	x				x			x		
Verbrauch				x	x	x	x			x								x	x		x		x			x		
Kilometer Stand	x			x			x	x										x	x				x					
Öl Stand	x										x								x				x					
Reifendruck	x						x						x	x	x	x							x					
Handbremse	x																											
Batterie Ladezustand			x		x					x											x		x		x	x		
Trips				x			x	x															x					
Warnung (z.b. Flüssigkeit)																							x					
Standort			x	x	x	x	x	x	x			x					x	x	x		x		x		x	x		
Manipulation (Intrusive)																												
Motor starten									x			x	x	x	x	x	x	x	x						x			
Türen öffnen/schließen	x		x	x	x	x	x		x	x		x	x	x	x	x	x	x	x			x			x			
Lichter ein/ausschalten	x			x	x	x			x				x	x	x	x	x	x							x			
Hupe	x			x	x	x							x	x	x	x	x	x							x			
Klimaanlage			x		x	x				x							x	x			x				x	x		
Alarmanlage									x									x										
Navigationsprogram	x	x								x							x									x		
Musik			x								x												x				x	
Service Termine	x	x																										
Einschränkungen (Geschw.)																			x			x						x
Verbindungstyp																												
Bluetooth/USB		x		x				x	x		x		x	x	x	x								x				
WLAN			x																				x				x	x
Internet	x	x			x	x	x		x	x		x	x	x	x	x	x	x	x		x	x			x	x		
Authentifizierung																												
FIN	x	x	x	x	x*	x*	x			x	x		x	x	x	x	x	x***	x	x	x	x			x	x		
PIN	x		x	x								x						x	x									
Autohändler							x		x								x			x		x			x			
Eindeutige ID										x		x																
Service-Account													x**	x**	x**	x**												
Bestätigung im Fahrzeug																					x****	x						
Anruf/Dokumente																										x		
Legende:	*	Only last 7 digits																										
	**	For Remote control																										
	***	Only last 8 digits																										
	****	Only to manage vehicle																										

Abbildung 1: Funktionsumfang von Apps populärer Autohersteller

Eine Analyse der verfügbaren Non-Intrusive Funktionen (siehe Abbildung 2) zeigt, dass eine Vielzahl der Hersteller das Auffinden des aktuellen Standort des Fahrzeuges via App ermöglichen. Dies soll ein einfaches Wiederfinden, beispielsweise in fremden Städten oder bei Car-Sharing Anwendungen, des Fahrzeuges ermöglichen. Ebenso geben viele Hersteller Auskunft über den aktuellen Ladezustand der Batterie bei Elektroautos respektive der Tank-Füllung des Fahrzeuges. In der Mehrzahl der Fälle kann diese Information auch über das Internet abgerufen werden, was einen Trend in diese Richtung vermuten lässt. Weitere verfügbare Funktionen sind das Abrufen von Flüssigkeitsständen, Warnungen sowie des allgemeinen Zustands des Fahrzeuges.

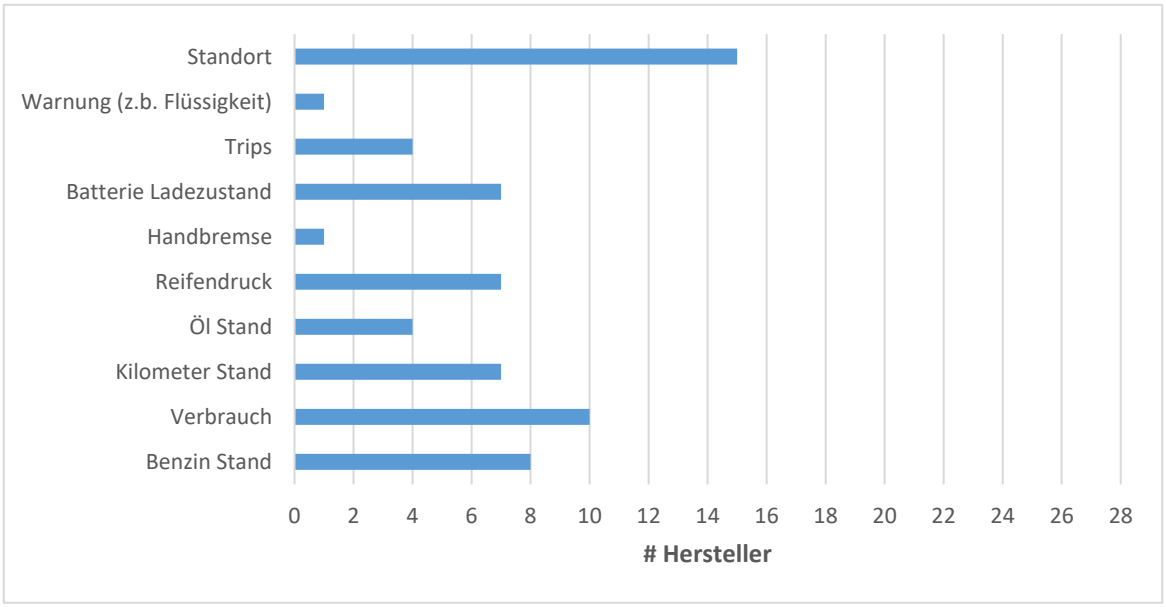

Abbildung 2: Übersicht aller Non-Intrusive Funktionen

Im Vergleich dazu gibt Abbildung 3 einen Überblick aller Intrusive Funktionen. Auffallend ist die häufige Implementierung der Remote-Funktion zum Öffnen bzw. Schließen der Türen. Auch im Falle von Intrusive Funktionen lässt sich ein klarer Trend in Richtung Kommunikation via Internet erkennen.

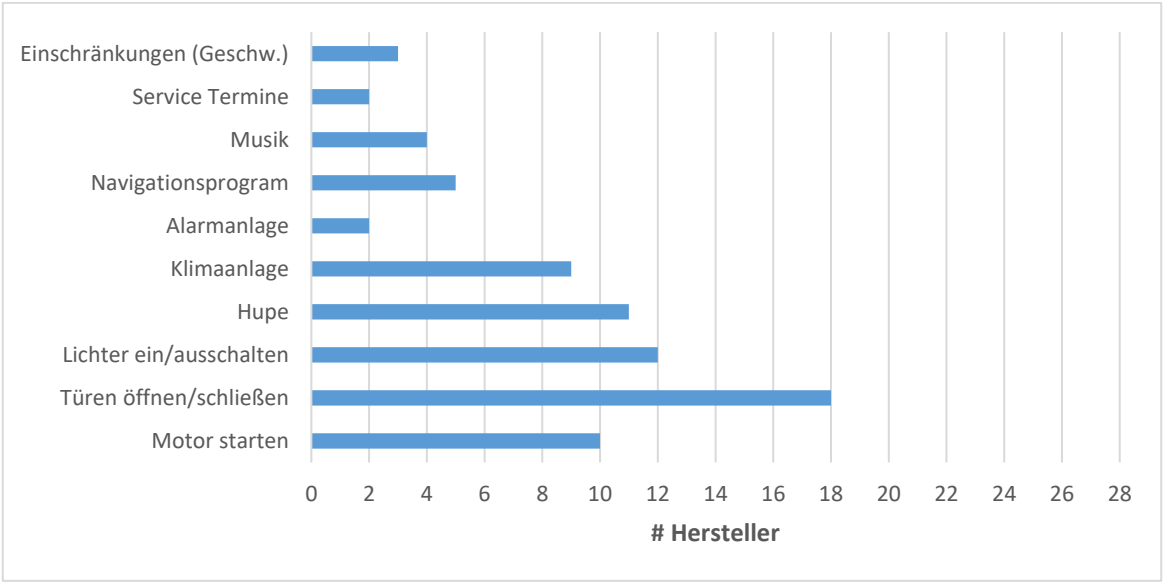

Abbildung 3: Übersicht aller Intrusive Funktionen

3.2. Authentifizierungsmechanismen

Um Benutzerinnen und Benutzer bzw. deren Fahrzeuge vor unautorisierten Zugriffen Dritter zu schützen, setzen Hersteller auf zusätzliche Authentifizierungsmechanismen, um deren mobile Begleitapplikationen mit dem Fahrzeug zu koppeln. Alle im Rahmen dieser Studie untersuchten Applikationen benötigen aus diesem Grund einen Benutzer-Account, um mit dem Fahrzeug über das Internet (falls verfügbar) zu kommunizieren. Als zusätzlicher Sicherheitsmechanismus wird ein zweiter Faktor eingesetzt, um das Fahrzeug mit diesem Account zu koppeln. Applikationen die nur über Bluetooth oder WLAN mit dem Fahrzeug in Verbindung stehen, benötigen in den meisten Fällen keinen zusätzlichen Authentifikationsfaktor.

Im Folgenden werden die beobachteten Authentifizierungsprozesse, also die Prozesse die notwendig sind um ein Fahrzeug mit einem User-Profil zu verlinken, beschrieben. Im Rahmen dieser Studie wurden sieben verschiedene Authentifizierungsmechanismen identifiziert, die entweder alleine oder in Kombination mit anderen Methoden zum Einsatz kommen.

3.2.1. FIN Authentifizierung

Eine FIN (Fahrzeug-Identifikationsnummer, engl. VIN) ist eine alphanumerische Zeichenfolge (ohne I, O und Q), die ein Fahrzeug eindeutig identifiziert [3]. Sie besteht im Grunde aus drei Komponenten: Einem dreistelligen Herstellercode, einem herstellerspezifischen Code (meist Fahrzeug-Typ) und einer fortlaufenden Seriennummer (siehe Abbildung 4). Die FIN ist aus Manipulationsgründen mehrfach am Fahrzeug angebracht und ist im Prinzip von jedem einsehbar (meist direkt an der Windschutzscheibe angebracht). Aufgrund dieser Eigenschaften dient die FIN nicht als eindeutiger Authentifizierungsfaktor, respektive Besitznachweis, und sollte lediglich dazu verwendet werden, um Fahrzeuge zu identifizieren oder die Applikation zu konfigurieren.

Herstellercode			Hersteller Schlüssel						Fortlaufende Nummer							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
W	B	Y	1	Z	2	1	0	4	0	V	Z	5	9	3	2	6

Abbildung 4: Aufbau einer Fahrzeug-Identifikationsnummer und Beispiel

3.2.2. PIN Authentifizierung

Bei Applikationen mit PIN Authentifizierung wird ein zusätzliches Geheimnis benötigt, um das Fahrzeug mit dem Benutzer-Account zu koppeln. In den meisten Fällen wird ein PIN-Code direkt im Auto angezeigt, oder vom Händler ausgefertigt. Dieser muss, oft in Kombination mit der FIN, beim Registrierungsprozess angegeben werden und ist in der Regel nur einmal gütig. In einigen Fällen muss diese PIN im Auto direkt eingegeben werden.

Je nach System bietet dieses Verfahren Vor- und Nachteile. Während ein zusätzlicher Faktor definitiv die Sicherheit fördert, bedeutet das gleichzeitig Einbußen in der Usability, da der PIN-Code erst vom Autohändler angefordert werden muss. Gleichzeitig beeinflusst die Länge der PIN bzw. der zusätzlich implementierten Sicherheitsmaßnahmen, wie Schutz vor Brute-Force Attacken oder Gültigkeit dieser PIN, die Sicherheit des Gesamtsystems. Bei zu kurzen PIN-Codes können diese durch einfaches ausprobieren aller möglichen Kombinationen erraten werden. Dies ist theoretisch ebenfalls möglich, falls die PIN im Fahrzeug direkt eingegeben werden muss, da die Kommunikation zwischen Fahrzeug und Server gegebenenfalls ebenfalls simuliert werden kann.

3.2.3. Authentifizierung via Autohändler

Bei einigen Systemen kann die Fahrzeug-Besitzerin oder der Fahrzeug-Besitzer die Remote Funktionalität seines Fahrzeuges selbst nicht direkt in Betrieb nehmen. Stattdessen muss zuvor ein autorisierter Autohändler das Fahrzeug im System eintragen bzw. mit dem Kunden-Account

verknüpfen. Der Prozess dafür variiert je nach Hersteller. Social-Engineering Angriffe könnten allerdings auch dieses System umgehen.

3.2.4. Eindeutige ID

Ähnlich wie bei PIN-Systemen wird hier als zusätzlicher Faktor eine eindeutige Zeichenfolge verlangt. Diese ist in den meisten Fällen länger als ein PIN-Code (z.B. Connectivity ID beim Anmelden zu Premium-Diensten bzw. SIM ID). Nachteil dieses Systems ist allerdings deren Gültigkeit. Während PINs oft nur einmal verwendet werden können, können diese eindeutigen IDs mehrfach für einen Registrierungsprozess verwendet werden.

3.2.5. Service Account

Viele Hersteller verlangen Service-Gebühren für das Verwenden der Remote-Funktionalität ihrer Fahrzeuge. Hierfür muss zuerst ein dementsprechender (Bezahl-) Account angelegt werden über diesen dann die Dienste genutzt werden können. Im Anschluss wird dieser Service-Account mit dem Benutzer-Account des Fahrzeug-Besitzers verlinkt. Zusätzlich muss man sich über den Service-Account im Auto direkt einloggen.

3.2.6. Bestätigung im Fahrzeug

Beim Verlinken des Fahrzeuges mit dem Benutzer-Account muss dieser Prozess im Fahrzeug explizit bestätigt werden. Im Vergleich zu PIN Authentifizierung findet hier allerdings kein Abgleich einer Verifikationsnummer statt.

3.2.7. Anruf oder Dokument als Besitznachweis

Ähnlich zur Authentifizierung via Autohändler wird bei diesem Prozess ein Anruf bei einer Authorisierungsstelle oder ein Dokument als Besitznachweis gefordert. Letzteres kann direkt über die Applikation als Foto hochgeladen werden.

4. Risiken

Durch die Einführung zusätzlicher Komponenten, wie Netzwerk-Funktionalität oder Hardware, werden Autos zunehmend intelligenter. Dies eröffnet neue Funktionen sowie Vorteile. Gleichzeitig zieht diese zunehmenden Technologisierung auch neue Bedrohungen nach sich und zwingt Hersteller auch alternative Angriffsszenarien, wie zum Beispiel Angriffe aus dem Internet, in Betracht zu ziehen [4]. Auf Grund der unterschiedlichen Natur der möglichen Schwachstellen wird in dieser Studie auf eine generelle Kategorisierung der Angriffe verzichtet. Sie beschränkt sich lediglich auf verschiedene Angriffsszenarien und mögliche Angriffe auf Smart-Vehicles und deren mobilen Begleitapplikationen ohne auf eine bestimmte Sortierung Rücksicht zu nehmen.

Im Allgemeinen können Angriffe verschiedene Auswirkungen haben. Sie können die Privatsphäre der Benutzer verletzen und so zum Beispiel den Standort des Autos/Benutzers oder im Benutzerprofil hinterlegte Daten hervorbringen. Gleichzeitig können Angriffe Schäden am Fahrzeug hervorrufen oder den Verlust der Kontrolle über dieses bewirken. Wir definieren diese Eigenschaften deswegen als: P: Privacy, C: Kontrolle des Fahrzeuges bzw. D: Denial of Service ((Remote-) Steuerung des Fahrzeuges nicht mehr möglich). Eine Übersicht identifizierter Eigenschaften gibt Tabelle 1.

Risiko	P	C	D
Umgehen des Authentifizierungsmechanismus	x	x	x
Auslesen/Erraten der Session-Information	x	x	
Benutzerprofil/Schwaches Passwort	x	x	x
Fehler im Protokoll	x	x	
Denial of Service			x
Physischer Zugang zum Gerät	x	x	

Tabelle 1: Identifizierte Risiken und deren Eigenschaften

Legende: P – Privacy, C – Kontrolle des Fahrzeugs, D – Denial of Service

4.1. Umgehen des Authentifizierungsmechanismus

Ein notwendiger Schritt beim Steuern eines Fahrzeuges via mobilen Apps ist die Verlinkung dieses mit dem Benutzerprofil. Je nach Implementierung (Kapitel 3.2) ergeben sich dabei unterschiedliche Möglichkeiten. Gelingt es einem Angreifer den Authentifizierungsprozess zu umgehen, ohne im tatsächlichen Besitz des Fahrzeuges zu sein, kann dieses einem beliebigen Benutzer-Account hinzugefügt werden. In Folge dessen können alle Funktionen des jeweiligen Herstellers, wie zum Beispiel das Abrufen von Statusinformationen über das Auto (inkl. Standort) oder die Steuerung von Komponenten, ohne jegliche Einschränkungen genutzt werden. Zusätzlich kann durch diesen Vorgang ein Fahrzeug aus einem vorhandenen Account gelöscht werden. Der Authentifizierungsmechanismus ist also ein kritischer Prozess jeder Applikation und bedarf gesonderter Absicherung.

Auf Grund oben genannter Eigenschaften definieren wir folgende Charakteristiken: P (Privacy), C (Kontrolle des Fahrzeugs), D (Denial of Service).

4.2. Auslesen/Erraten der Session-Information

Ein wichtiger Bestandteil einer Applikation ist das Session Management. Üblicherweise wird beim Login Vorgang vom Server ein eindeutiger Session-Token generiert, der in weiterer Folge bei jedem Request vom Client an den Server mitgeschickt wird. Dies erlaubt dem Server die eindeutige Identifizierung eines Users, ohne dass sich dieser beim Nutzen der Applikation erneut authentifizieren muss. Im Allgemeinen haben Session-Tokens jedoch eine begrenzte Gültigkeitsdauer.

Ein Angreifer hat mehrere Möglichkeiten einen Session-Token abzugreifen: beim Login Vorgang etwa durch einen Man in the Middle (MiM) Angriff oder wenn die Session aus der App, beispielsweise durch Malware, ausgelesen werden kann. Gelingt es einen Session-Token auszulesen, kann man die Identität einer Person stehlen und somit alle Funktionen der mobilen Applikation nutzen. Deswegen können folgende Eigenschaften festgestellt werden: P (Privacy), C (Kontrolle des Fahrzeugs).

4.3. Benutzerprofil/Schwaches Passwort

Schwache Passwörter sind nach wie vor ein Problem bei vielen Webdiensten. Viele User neigen dazu, einfache Passwörter zu verwenden, diese ungesichert abzuspeichern oder wiederzuverwenden [5]. Vorfälle in jüngster Vergangenheit zeigen, dass Accountdaten auch durch Verfehlungen bei der Datensicherheit großer Dienste gestohlen werden können [6].

Eine Übernahme des Accounts bedeutet im Fall von Applikationen zur Steuerung von Smart-Vehicels automatisch Zugriff auf das Fahrzeug beziehungsweise alle Funktionen die die App zur Verfügung stellt. Daraus ergeben sich nachfolgende Eigenschafen; P (Privacy), C (Kontrolle des Fahrzeugs). Kann durch die Übernahme des Accounts zusätzlich noch das Passwort geändert werden, so kann auch die Charakteristik D (Denial of Service) zugewiesen werden.

4.4. Fehler im Protokoll

Erst kürzlich wurde gezeigt, dass es möglich ist durch fehlerhafte Kommunikationsprotokolle (z.B. fehlende Absicherung nach Außen) die Kontrolle über Fahrzeugen ohne großen Aufwand zu erlangen [1]. Es wurde gezeigt, dass eine Authentifizierung basierend auf der FIN keinen ausreichenden Schutz bieten kann. Als Resultat kann ein Angreifer ohne Authentifizierung die Kontrolle über beliebige Fahrzeuge gewinnen.

Somit ergeben sich als Charakteristiken P (Privacy), C (Kontrolle des Fahrzeugs) .

4.5.Denial of Service

Viele Applikationen bieten die Möglichkeit remote das Fahrzeug zu entsperren oder den Motor zu starten/stoppen. Eine App kann also als Schlüsselersatz dienen.

Gezielte Angriffe auf diese Remote-Funktionalität können dazu führen, dass das Auto nicht mehr entsperrt oder gestartet werden kann. Ein Beispiel dafür ist das (beabsichtige) Hervorrufen einer Accountsperre durch zu häufige Eingabe eines falschen Passworts. Denial of Service Attacken können aber darauf abzielen, die Benutzung des Fahrzeuges, beispielsweise durch eine dauerhafte Aktivierung der Klimaanlage und die damit einhergehende Entleerung der Batterie, zu verhindern.

In den meisten Fällen kann durch Denial of Service Attacken jedoch kein direkter Zugriff auf den Benutzeraccount erlangt werden. Folglich ergibt sich die Eigenschaft D (Denial of Service).

4.6. Physischer Zugang zum Gerät

In gewissen Szenarien kann es einem Angreifer, entweder durch Diebstahl, Verlust oder anderen komplexen Methoden, gelingen, vollen Zugang zu einem Smartphone zu bekommen. Unter der weiteren Annahme, dass dieser Angreifer grundlegendes Wissen im Bereich Kryptographie sowie der Analyse von mobilen Applikationen hat, kann diese Person beliebige Daten (wie z.B. Session-Information) aus den Applikationen extrahieren beziehungsweise deren Funktionsumfang nutzen. In Kombination mit Standort-Diensten kann somit im schlimmsten Fall das Fahrzeug entwendet werden.

Der Verlust eines Mobiltelefons kann den gleichzeitigen Verlust der Kontrolle über das Fahrzeug bedeuten. Die Eigenschaften P (Privacy) und C (Kontrolle des Fahrzeugs) können zugeteilt werden.

4.7. Schutzmechanismen

Die unterschiedliche Natur möglicher Angriffe gegen mobile Applikationen für Smart-Vehicles stellt eine Herausforderung dar, eine generelle Aussage zu treffen. Vielmehr sollen die hier diskutierten Schutzmechanismen als Hilfestellung gesehen werden, um mögliche Angriffe zu unterbinden oder diese zu erschweren.

Wir haben gezeigt, dass der Authentifizierungsmechanismus ein kritischer Prozess ist. Der verwendete Mechanismus muss also eindeutig sein und darf nicht an ein öffentliches „Geheimnis“ wie eine FIN gebunden sein. Ebenso sollten Maßnahmen getroffen werden, dass ein nachträgliches Koppeln des Fahrzeuges mit einem zweiten Account nicht möglich ist. Brute-Force Attacken gegen Einmal-PINs können durch eine Limitierung der möglichen Versuche unterbunden werden.

In Bezug auf die Gerätelandschaft können Automobilhersteller nicht davon ausgehen, dass Smartphones mit einem Passwortschutz, Geräteverschlüsselung oder HSM ausgestattet sind. Folglich sind zusätzliche Maßnahmen notwendig, um den Session-Token zu schützen. Dies kann beispielsweise durch einen vom User gewählten PIN-Code realisiert werden. Mittels einer Key-Derivation-Funktion kann im Anschluss der Schlüssel für die Verschlüsselung des Session-Token generiert werden. Ebenso kann die Sicherheit durch Setzen einer Gültigkeitsdauer für die Session erhöht werden.

Die Analyse von Applikationen hat gezeigt, dass schwache Passwörter eine Gefahr darstellen. Stärkere Passwort-Richtlinien wie zum Beispiel verpflichtende Sonderzeichen sind eine Möglichkeit, der Gefahr vorzubeugen, bei virtuellen Tastaturen von Smartphones aus der Anzahl der komfortabel eingebbaren Sonderzeichen eingeschränkt. Um gleichzeitig andere Dienste besser zu schützen, etwa bei Wiederverwendung von Passwörtern, sollte beim Login Vorgang das Passwort nicht in Plain-Text geschickt werden, sondern durch eine Einwegfunktion für einen Angreifer unkenntlich gemacht werden. Zusätzliche Sicherheit bietet außerdem die Verwendung einer „2-Schritt-

Verifikation“. Durch Hinzufügen eines zweiten Faktors zum Login Prozess kann die Accountübernahme im Falle eines Datenschutzproblems erschwert werden.

5. Analyse

Jüngste Vorfälle haben gezeigt, dass einige Applikationen durchaus Schwachstellen und gravierende Sicherheitslücken offenbaren. Um einen Überblick über die Situation zu erhalten, untersuchen wir populäre Apps im Detail, um potentielle Sicherheitslücken zu identifizieren. Ziel ist es „Bad Practices“ zu identifizieren bzw. beobachtete Design-Entscheidungen und mögliche Auswirkungen zu dokumentieren.

Dafür wurden die Android Applikationen in Abbildung 1 dekompiliert und deren Dalvik–Code analysiert. In unserem Szenario setzen wir außerdem voraus, dass ein potentieller Angreifer vollen Zugang zu einem Smartphone ohne aktivierte Plattform-Verschlüsselung erhält und Android-Applikationen analysieren sowie das Dateisystem auslesen kann.

5.1. Authentifizierungsprozess

Gelingt es einem Angreifer, den Authentifizierungsmechanismus einer App zu umgehen, kann der volle Funktionsumfang der App genutzt werden. In den beobachteten Applikationen konnte der Authentifizierungsmechanismus keiner Applikation ohne gezielte Attacken (Bsp. Social Engineering, Brute-Force) umgangen werden. Gezielte Angriffe würden über den Rahmen dieser Arbeit hinausgehen.

Allerdings wurde in einigen Fällen festgestellt, dass ein Denial of Service Angriff auf gewisse Authentifizierungsprozesse möglich ist. Bei gewissen Herstellern genügt es die FIN zu einem Benutzerprofil hinzuzufügen um dieses aus einem bestehenden Account zu löschen und somit die Remote Funktionalität eines bereits registrierten Fahrzeuges zu deaktivieren. Folglich ist eine erneute Authentifizierung notwendig. Verlässt sich eine Benutzerin oder ein Benutzer auf die Remote-Funktionalität (z.B. Entsperren des Fahrzeuges) via Smartphone, kann dieses ohne passenden Schlüssel nicht mehr verwendet werden.

5.2. Credential Storage

Um applikationsspezifische Daten unter Android abzulegen, gibt es mehrere Möglichkeiten, wie zum Beispiel *SharedPreferences*, *Datenbank* oder das *File System*. In den meisten beobachteten Fällen wird auf die SharedPreferences[2] zurückgegriffen, um Session Informationen abzulegen. Diese dienen dazu, einfache Key-Value Paare persistent zu speichern und werden unter Android in einer XML-Datei am Dateisystem am Gerät gesichert. Standardmäßig befindet sich die Datei im Applikationsverzeichnis und unterliegt somit den gleichen Datei-Berechtigungen. Auf nicht gerooteten Geräten[3] kann folglich nur die Applikation selbst dieses Dokument auslesen.

Erlangt ein Angreifer Vollzugriff auf die Datei, beispielsweise durch „rooten“ des Geräts, kann deren Inhalt gelesen werden, da keine zusätzlichen Schutzmaßnahmen wie Verschlüsselung aktiv sind. Zwar bietet Android seit Version 2.3 die Möglichkeit einer Plattform-Verschlüsselung, allerdings können Applikations-Entwickler nicht davon ausgehen, dass diese aktiv ist oder dass das Gerät keine Modifikationen am Berechtigungssystem erfahren hat.

Um sensible Daten sicher in den SharedPreferences abzulegen sind zusätzliche Schutzmaßnahmen notwendig. Im Folgenden werden die beobachteten Praktiken diskutiert.

[2] https://developer.android.com/reference/android/content/SharedPreferences.html
[3] Geräte ohne “Root”/Admin-Account

5.2.1. Keine Verschlüsselung

Werden Session Informationen ohne Verschlüsselung abgelegt, bietet die Applikation bei gerooteten Geräten oder Geräten ohne aktivierter Plattform-Verschlüsselung nicht ausreichend Schutz. In Folge dessen kann ein Angreifer, der Vollzugriff auf ein Gerät hat, beliebige Daten aus den Shared-Preferences oder dem Dateisystem auslesen.

Auswirkung: Selbst bei zusätzlichen Maßnahmen wie Passwort-Schutz der Applikation kann die Session-Information ausgelesen werden und somit die zur Verfügung gestellte Funktionalität genutzt werden.

5.2.2. Statischer Initialisierungsvektor

Um zu verhindern, dass gleiche Nachrichtenblöcke bei Einsatz von Blockchiffren denselben Geheimtext produzieren, wird für das Verschlüsseln eine zufällige Zeichenkette als weiterer Parameter gewählt. Dieser Initialisierungsvektor sollte für jede Nachricht neu generiert werden und ist im Allgemeinen nicht geheim.

Die Analyse ergab, dass ein Teil der untersuchten Applikationen diesen Parameter statisch in der Applikation definieren und nicht für jede Operation neu generieren.

Auswirkung: Ein statischer Initialisierungsvektor hat zur Folge, dass gleiche Nachrichten denselben Geheimtext produzieren. Dies lässt einen Angreifer Rückschlüsse auf die verschlüsselten Daten ziehen.

5.2.3. Statischer/Remote Key

Bei der Verschlüsselung der Session Information muss sichergestellt werden, dass der dafür verwendete Schlüssel sicher verwahrt ist. Auf Geräten ohne Hardware-Security-Module (HSM) kann dies im Allgemeinen nicht bewerkstelligt werden. Ein Lösungsansatz dafür ist das Verwenden von Key-Derivation Funktionen. Dabei wird der Schlüssel on demand aus einem von der Benutzerin oder dem Benutzer definierten Passwort abgeleitet.
In den untersuchten Applikationen wurde festgestellt, dass der Schlüssel teilweise fix in der App festgelegt wurde oder beim Starten der Applikation von einem Server geladen wurde.

Auswirkung: Ein statischer/remote Schlüssel bietet praktisch keinen Schutz, da der Schlüssel von einem Angreifer ausgelesen werden kann. Die verschlüsselten Daten können folglich ohne Aufwand entschlüsselt werden.

5.2.4. Weak Algorithm

Verschlüsselung bietet im Allgemeinen nur so viel Schutz wie der zu Grunde liegende Algorithmus. In den letzten Jahren wurden mehrere erfolgreiche Angriffe auf verschiedene Algorithmen präsentiert. E. Barker [7] gibt dazu eine Empfehlung über starke Verschlüsselungsverfahren ab.

Im Rahmen der Studie wurden Applikationen identifiziert, die schwache Verschlüsselungs-algorithmen verwenden.

Auswirkung: Schwache Algorithmen können nicht ausreichend Schutz (Authentication, Non-Repudiation, Confidentiality) für sensible Daten bieten.

5.3. API Protection

Ein (Web) Application Programming Interface ist ein Dienst durch das Applikationen mit einem Service interagieren können. Absicherung der API gegen Angriffe ist essentiell um einen Service garantieren zu können.

Die untersuchten Applikationen und deren APIs bieten zum Teil nicht genügend Schutz gegenüber Brute Force Attacken. Weiters wurde im Zuge der Analyse festgestellt, dass bei mehreren Applikationen kein Certificate Pinning, ein Prozess in dem die Applikation das Zertifikat des Servers verifiziert, zum Einsatz kommt.

Auswirkung: Durch den fehlenden Brute Force Schutz von APIs ist es möglich, beliebig viele Anfragen ohne Restriktion an einen Server zu schicken. Somit können durch ausprobieren aller Möglichkeiten Parameter erraten werden (Bsp. PIN). Außerdem kann ein Angreifer, der Vollzugriff auf ein Gerät hat, eigene Zertifikate importieren, und durch das fehlende Certificate Pinning die Kommunikation der Applikation mitlesen.

5.4. Miscellaneous

Im Zuge der Analyse der Applikationen wurden Implementierungsdetails und Praktiken gefunden, die sich keiner allgemeinen Kategorie zuordnen lassen. Im Folgenden wird deswegen eine Sammlung an beobachteten Details diskutiert, die Angriffe ermöglichen können.

5.4.1. Zufallszahlen

Zufallszahlen bilden die Grundlage der meisten sicheren Systeme. Sie werden beispielsweise dazu verwendet, nach erfolgreicher Authentifizierung eine zufällige Session ID zu generieren. Zu Grunde liegt dabei häufig ein Pseudo-Zufallszahlen-Generator. Wichtig dabei ist es, dass es unmöglich ist aus einer gegebenen Sequenz an Zufallszahlen, die zukünftigen Werte vorherzugsagen.

In einer der untersuchten Apps wurde `java.util.Random` verwendet um Session-Informationen zu erzeugen. Diese Klasse erfüllt allerdings nicht die gewünschten Eigenschaften. Somit kann ein Angreifer Teile dieser Session Informationen vorhersagen.

5.4.2. Passwort-Verschlüsselung

Ein kritischer Aspekt jeder Applikation ist der Authentifizierungsprozess. Das Übermitteln der Zugangsdaten in Plain-Text (trotz verschlüsselter Verbindung) birgt potentielle Probleme. Kann ein Angreifer die Kommunikation abhören, bedeutet das gleichzeitig, dass die Zugangsdaten kompromittiert werden.

In einer der untersuchten Applikationen wurde das Passwort symmetrisch mit einem Schlüssel, der zuvor vom Server geladen wurde verschlüsselt. Allerdings können Angreifer die den Verkehr mitlesen können, diesen übermittelten Schlüssel auslesen und folglich auch das verschlüsselte Passwort entschlüsseln. Diese Maßnahme bietet also keinen zusätzlichen Schutz.

5.4.3. Zugangsdaten in Plain Text

Bei einigen Applikationen wurde beobachtet, dass nicht nur die Session Information, sondern auch die eingegebenen Daten unverschlüsselt in den SharedPreferences abgelegt werden. Angreifer, die Zugriff auf diese Datei bekommen, können somit den Benutzernamen als auch das Passwort auslesen.

6. Schlussfolgerung

Die Kombination von Automobilen mit modernen Kommunikationstechnologien eröffnet neue Möglichkeiten. Durch Smartphone Anwendungen ist es möglich, aus der Ferne Autos zu entsperren, den Motor zu starten oder Statusinformationen über das Fahrzeug abzurufen. Allerdings ziehen diese Funktionen neue Risiken mit sich. Komponenten, die ursprünglich nicht dafür entwickelt wurden, müssen nun gegen den Zugriff von außen oder Fremden abgesichert werden.

Im Fokus der Studie stand die Untersuchung von populären Applikationen. Im Zuge dessen wurden mobile Apps von Automobilherstellern analysiert und deren Funktionsumfang dokumentiert. Die Studie gibt außerdem eine Übersicht über die möglichen Steuerungsmöglichkeiten und teilt die Funktionen in zwei Kategorien: „Intrusive“ und „Non Intrusive“. Diese Kategorisierung dient als Basis für weitere Untersuchungen. Die vorliegende Studie befasst sich außerdem mit möglichen Risiken sowie mögliche Schutzmechanismen, die durch die Einführung mobiler Applikationen im Fahrzeug Bereich entstehen können. Abschließend wurden Applikationen im Detail analysiert und potentielle Angriffsvektoren sowie Designentscheidungen identifiziert.

Die erzielten Ergebnisse zeigen, dass in vielen Fällen Implementierungsfehler oder schwache Designentscheidungen negative Auswirkungen auf den Sicherheitslevel der Applikationen haben.

Die Analyse zeigt außerdem, dass der Trend klar in Richtung Einsatz von Kommunikations-technologien und mobilen Anwendungen im Fahrzeugbereich geht.

7. Abbildungsverzeichnis

8. Literaturverzeichnis

[1] T. Hunt, *Controlling vehicle features of Nissan LEAFs across the globe via vulnerable APIs,* 2016.

[2] S. Ma, „The Solution of an IoT Application: Smart Vehicle,“ *Iccta,* 2011.

[3] 2. ISO/TC, *ISO 3779:2009,* 2009, p. 6.

[4] A. Humayed und B. Luo, „Cyber-Physical Security for Smart Cars,“ *ACM/IEEE 6th International Conference on Cyber-Physical Systems (ICCPS),* pp. 252-253, 2015.

[5] J. Yan, B. Alan, R. Anderson und A. Grant, „Password memorability and security: Empirical results,“ *IEEE Security and Privacy,* Bd. 2, Nr. 5, pp. 25-31, 2004.

[6] C. (. Scott, *Protecting Our Members,* 2016.

[7] E. Barker, „Recommendation for Key Management Part 1: General,“ Gaithersburg, 2016.