A=SIT Technology
E-Government Innovationszentrum

6. April 2021 Studie · Transparenz · Web & Cloud · 26 Min. Lesezeit

Neue Vertrauensmodelle durch Trusted Computing

Archiviert. Dieser Inhalt wird nicht mehr aktualisiert. Er bleibt zur Information verfügbar.

Stand 2021-03-26
Quelle https://technology2.a-sit.at/veroeffentlichungen/neue-vertrauensmodelle-durch-trusted-computing/

Traditionell gibt es zwei grundlegende Ansätze, um Vertrauensmodelle zu implementieren. Einerseits gibt es mit PKIX eine hierarchische Vertrauensstruktur mit klar zugewiesenen Rollen, andererseits wird mit Web-of-Trust-Konzepten eine gegenläufige Strategie ohne zentrale Instanzen verfolgt. Durch die zunehmende Verfügbarkeit von Trusted Computing (beispielsweise in Form von Software Guard Extensions, oder auf Mobilgeräten mit kryptografischer Hardware) ergeben sich potentiell Möglichkeiten beide Modelle miteinander zu kombinieren. Im Rahmen dieses Projekts wurden solche Möglichkeiten einer Vereinigung dieser konträren Vertrauensmodelle erforscht. Im Zuge dessen wurde ein hybrides Vertrauensmodell entwickelt, welches mittels Trusted Computing die propagierten Vorteile eines Web of Trust auch auf globalen Skalen nutzbar macht, ohne dass dabei die diesem Konzept als inhärent nachgesagten Nachteile schlagend werden.

1. Einleitung

Vertrauensmodelle wurden traditionell technologieunabhängig definiert. Klassische, hierarchisch abgebildete Vertrauensverhältnisse sind zwar im Rahmen von PKIX über Zertifikate durch die *Public Key Infrastructure (X.509) Working Group* [1] auch technisch standardisiert, die grundlegenden Annahmen und Prinzipien lassen sich jedoch unabhängig von konkreten Implementierungen anwenden. Dies wird auch durch die Evolution eines anderen Standards, TLS in der Version 1.3 [1], verdeutlicht: Erstmals wird der direkte Einsatz von Public Keys zum Aufbau einer sicheren Verbindung unterstützt, ohne dass X.509-Zertifikatsketten zum Einsatz kommen müssen. Als eine Art Gegenpol zu X.509 hat sich *Pretty Good Privacy* (PGP) [2] innerhalb bestimmter Szenarien etabliert und kommt ohne PKIX-Strukturen wie Zertifizierungsstellen und vorkonfigurierten Vertrauensankern aus. Stattdessen wird auf ein so genanntes *Web of Trust* gesetzt; ein organisch wachsendes, nutzerzentrisches Netzwerk von kryptografischen Schlüsseln. Hierbei wurde von Anfang an gänzlich auf Zertifikate verzichtet. Statt wie im Rahmen von PKIX typischerweise den Herstellern von Software dahingehend zu vertrauen, dass Trust Stores mit einer Reihe vertrauenswürdiger Zertifizierungsstellen vorkonfiguriert ausgeliefert werden (und diese Stellen zu gewissen Graden dafür bürgen, Zertifikate nur an legitime Repräsentanten von Identitäten auszustellen),

steht jeder einzelne Nutzer und jede einzelne Nutzerin im Zentrum seines, bzw. ihres Vertrauensnetzwerks. Hierfür wurden im Sinne der Interoperabilität ebenfalls Protokolle, Dateiformate und Semantiken definiert. Die Grundlegenden Konzepte sind jedoch wie im Rahmen von PKIX-Hierarchien technologieunabhängig.

Das Web-of-Trust-Modell scheint sich auf Grund seiner inhärent dezentralen und von jeglichen strukturellen Regulatorien und hierarchischen Strukturen entkoppelten Charakteristika auf den ersten Blick insbesondere zur Vertrauensbildung, bzw. der Formalisierung von Vertrauensverhältnissen im Rahmen dezentraler Strukturen zu eignen. Tatsächlich wurde dem Webof-Trust-Ansatz jedoch bereits vor Jahrzehnten eine Tauglichkeit bestenfalls im Rahmen scharf abgegrenzter Domänen, keinesfalls jedoch als global einsetzbares Vertrauensmodell attestiert [2]. Gleichzeitig spielt die Vertrauensbildung in dezentral organisierten verteilten Systemen mangels einer zentralen Instanz, welche korrektes Verhalten unter den Akteuren eines solchen Systems durchsetzen könnte, eine tragende Rolle. Wie bereits in vorangegangenen A-SIT-Projekten erarbeitet, ergeben sich durch jüngere Entwicklungen im Trusted-Computing-Bereich in der Domäne der dezentral organisierten verteilten Systeme neue Möglichkeiten, bekannte Probleme zu lösen und vorhandene Angriffsvektoren zu entschärfen [3]. Dieses Projekt verfolgt diesen Ansatz konsequent weiter und wendet Trusted Computing im Kontext eines global einsetzbaren Vertrauensmodells an, um so die grundlegenden Prinzipien eines Web of Trust auch tatsächlich praktikabel umsetzen zu können. Die Basis dafür bildet die Möglichkeit, moderne Android-Smartphones als Trusted-Computing-Basis einzusetzen.

Dieses Dokument ist wie folgt strukturiert: Nachfolgend werden die grundlegenden Gesichtspunkte, unter denen Android als Trusted-Computing-Basis eingesetzt werden kann, sowie die generellen Eigenschaften dieses Konzepts zusammengefasst. Darauf aufbauend werden die Anforderungen an ein flexibles, global einsetzbares Vertrauensmodell abgeleitet. Anschließend werden die Charakteristika von hierarchischen Vertrauensmodellen einem Web of Trust gegenübergestellt. Auf Basis dieser Hintergrundinformationen wird schließlich ein neuartiges Vertrauensmodell vorgestellt, welches auf Basis von Trusted Computing die charakteristischen Eigenschaften eines Web of Trust auf globaler Ebene praktisch nutzbar macht.

2. Trusted Computing im dezentralen Kontext am Beispiel Android

Der Kern des Themenkomplexes rund um den Einsatz mobiler Plattformen als Trusted-Computing-Basis wurde bereits in vorangegangenen A-SIT-Projekten mehrfach [4] und aus unterschiedlichen Perspektiven [3] beleuchtet. Aus diesem Grund fassen die nachfolgenden zwei Unterabschnitte lediglich die im Kontext dieses Projekts relevanten Aspekte zusammen.

1 https://tools.ietf.org/wg/pkix/ 2 https://www.openpgp.org/

2.1. Sicherheitsmodelle mobil und am Desktop

Das Sicherheitsmodell von Mobilplattformen unterscheidet sich grundlegend vom Desktop-Bereich. Dies ist dem Umstand geschuldet, dass sowohl Apples iOS, als auch Googles Android-Mobilbetriebssystem keine bestehenden Ökosysteme bedienen mussten und daher neue, umfassendere und striktere Sicherheitsmodelle einführen könnten, ohne dass dies auf Grund von Kompatibilitätsproblemen bei einer vorhandenen Entwicklerbasis auf Ablehnung hätte stoßen können. Nach und nach wurden striktes Sandboxing von Applikationen und streng geregelter benutzerseitiger Zugriff auch auf das eigene Gerät zunehmend um Trusted-Computing-Konzepte wie verifizierte Bootvorgänge bis hin zu signierten Betriebssystemabbildern erweitert. Im krassen Gegensatz dazu war im Desktop-Bereich

Abwärtskompatibilität insbesondere auf konzeptioneller Ebene eine Anforderung, um im Markt bestehen zu können. Spätestens seit dem Siegeszug des IBM-kompatiblem PCs im Heimbereich war es Gang und Gäbe, Betriebssysteme nach Wahl installieren zu können und das Verhalten und Aussehen einer Desktop-Umgebung frei nach Nutzerwünschen anpassen zu können. Ebenso war die Hardware von Anfang an modular aufgebaut und über genormte, bzw. de-facto-standardisierte Schnittstellen ein Austausch von Hardware-Komponenten vorgesehen, wodurch sich der eigene Rechner erweitern und aufrüsten lies. Diese Grundprinzipien gelten nach wie vor und sind je nach Zielgruppe ausschlaggebend für den Verkaufserfolg von PCs und Laptops. Freie Betriebssysteme wie diverse Linux-Distributionen oder BSD-Derivate erfordern für die Installation zwingenden Vollzugriff auf Firmware-Interface (UEFI) und Festplatten um parallel zu oder anstatt eines vorinstallierten Betriebssystems installiert werden zu können. Eine derart offene Plattform kann folglich in weiten Teilen lediglich von der Benutzerin bzw. dem Benutzer als vertrauenswürdig erachtet werden. Aus Sicht Dritter kann jedoch weder ein Vertrauensverhältnis zum Betriebssystem noch einzelnen Applikationen hergestellt werden. Entsprechend wird Trusted Computing am Desktop „am Betriebssystem vorbei“ umgesetzt und trusted IO lässt sich nur schwierig und unter scharf abgegrenzten Bedingungen auf bestimmten Hardwarekonfigurationen umsetzen, sodass dieser Aspekt aus dem Trusted-Computing-Themenkomplex nicht als am Desktop verfügbar angesehen werden kann. Darüber hinaus wird im Desktop-Bereich eine Isolation von Anwendungen zueinander zwar vorangetrieben, tatsächlich ist es Stand 2021 jedoch nach wie vor üblich, dass eine Applikation, welche am System ausgeführt wird, auf die Daten anderer Applikationen, sowie auch auf persönliche Daten zugreifen kann. Dieses Verhalten wird im Allgemeinen von Nutzerinnen und Nutzern auch erwartet und eine diesbezügliche Restriktion würde höchstwahrscheinlich auf Ablehnung stoßen, da etablierte Arbeitsabläufe angepasst werden müssten.

Im Mobilbereich hingegen ist jeglicher Zugriff auf Dateisystem, Sensoren und Nutzerdaten über ein restriktives Berechtigungssystem reguliert und hardwaregestützte verifizierte Bootketten verhindern Systemmodifikationen. Insgesamt ergeben sich dadurch enorme Sicherheitsgewinne, auch wenn Besitzerinnen und Besitzer von Geräten dafür insofern in ihrer Verfügung über die eigenen Geräte eingeschränkt werden, als das lediglich vom Hersteller ausgelieferte Betriebssystemabbilder installiert werden können. Zwar erlauben einige Gerätehersteller das Entsperren von Geräte-Bootloadern und die anschließende Installation beliebiger Betriebssysteme, allerdings lässt sich dies per Remote Attestation feststellen, wodurch keine Vertrauensbasis zu derartig modifizierten Geräten mehr hergestellt werden kann. Details hierzu werden im nachfolgenden Abschnitt erläutert.

2.2. Remote-Attestation und Trusted Computing unter Android

Der Inhalt dieses Abschnitt ist eine Replik der Zusammenfassung eines vorangegangenen A-SIT- Projekts, die ursprünglich 2020 veröffentlich wurde [5] . Unter Android ist Trusted Computing inklusive trusted IO nicht nur aus Nutzersicht möglich, sondern auch Entwicklern und Entwicklerinnen von Applikationen, bzw. Servicebetreibern zugänglich. In einem vorangegangenen A-SIT-Projekt wurde ein entsprechendes Konzept erarbeitet und praktisch umgesetzt, das es ermöglicht, die Authentizität und Integrität einer Applikation aus der Ferne zu überprüfen [4]. Dabei wird ein Mobilgerät als Gesamtsystem in den Status einer Trusted-Computing-Basis angehoben. Hierfür wird eine Kombination aus am Gerät vorhandener kryptografischer Hardware zur Verwaltung kryptografischer Schlüssel, verifiziertem Bootvorgang und diverser Eigenschaften des Android-Sicherheitsmodells [6] in Verbindung mit Remote Attestation, der Integritätsprüfung aus der Ferne, eingesetzt. Die technische Basis für diese Umsetzung von Trusted Computing unter Android in dieser Form ist dem entsprechenden Projektbericht [7] zu entnehmen, weshalb an dieser Stelle lediglich ein Überblick über dieses Verfahren gegeben wird. Im Kern basiert das Verfahren

darauf, dass in vertrauenswürdigen Hardwarekomponenten am Smartphone vom Hersteller ein public-private Keypair, sowie eine zugehörige Zertifikatskette hinterlegt werden. Die Wurzel dieser Kette ist ein Google-Root-Zertifikat. Darüber hinaus sind Informationen über Widerrufspunkte (wenn auch nicht CRL und OCSP-konform) in dieses Zertifikat kodiert. Beim Erstellen eines symmetrischen Schlüssels oder eines asymmetrischen Schlüssel-paares aus einer Smartphone-Applikation kann optional eine kryptografische Nonce, die so genannte Attestation Challenge, mitgegeben werden. Wenn vorhanden, fließt dieser Parameter in die Generierung eines Zertifikats ein, welches in der Hardware vom vom Hersteller hinterlegten private Key signiert wird. Dadurch entsteht eine Zertifikatskette, welche den neu erstellten Schlüssel, bzw. das Schlüsselpaar zertifiziert. Über zusätzliche X.509-Zertifikatserweiterungen werden Zustand, bzw. Integrität des Bootloaders, des Betriebssystems und der Applikation im Zertifikat hinterlegt. Eine entsprechende Auswertung dieser Erweiterungen, sowie Validierung der Zertifikatskette ermöglichen es, die Integrität des Geräts, Betriebssystems, sowie der Applikation zu verifizieren. Dadurch können auch Modifikationen durch den Benutzer bzw. die Benutzerin ausgeschlossen, bzw. erkannt werden. Folglich ist es Applikationsentwicklern bzw. Applikations-entwicklerinnen und Service-Providern möglich, die volle Kontrolle über installierte Applikationen zu behalten und auch sensible Operationen auf Android-Smartphones auszulagern.

Voraussetzung für dieses Verfahren ist jedoch das Vorhandensein entsprechender Hardware-Module im Smartphone, was für faktisch alle seit Mitte 2019 erschienenen Geräte der Fall ist. Durch die Auswertung der in Zertifikatserweiterungen kodierten Parameter wie Betriebssystemversion, Patch-Level, usw. ist es möglich, ein Vertrauensniveau zu attestieren. Ein nicht aktuelles Betriebssystem kann als weniger vertrauenswürdig angesehen werden als ein aktuelles. Wird beispielsweise ein entsperrter Bootloader detektiert, so senkt dies das Vertrauensniveau drastisch, da in diesem Fall Modifikationen am Betriebssystem nicht ausgeschlossen werden können. Sofern ein Smartphone verwendet wird, das schlicht zu alt ist und die notwendige Funktionalität nicht unterstützt, kann keine Vertrauensbasis aufgebaut werden. Dieser Fall wird jedoch zunehmend irrelevant, da derartige Geräte kaum noch vertrieben werden.

2.3. Generalisierte Eigenschaften des Android-basierten Trusted-Computing- Konzepts Android zeichnet sich als Trusted-Computing-Basis gleich mehrfach von konkurrierenden Konzepten und Systemen ab. Insbesondere die Tatsache, dass Android eben nicht explizit für den Einsatz im Trusted-Computing-Kontext konzipiert wurde und den entsprechenden Anspruch auch nicht erhebt, geht mit einem Maß an Flexibilität einher, der eine Einbettung in Web-of-Trust-artige Vertrauenskonzepte ermöglicht (Details hierzu werden in Abschnitt 4 dargelegt). Entsprechend wird ein Vertrauensverhältnis nicht binär sondern graduell formuliert. Generell können folgende Charakteristika für Android-basiertes Trusted Computing abgeleitet werden:

1. *Niederschwellige Zugänglichkeit* : Zum Zeitpunkt der Einführung im Jahr 2019 war der Vertrauensaufbau wie unter Android mittels Remote Attestation möglich, einzigartig. Die daraus resultierende Konsequenz, de-facto unveränderte Programme, welche weiterhin uneingeschränkten Zugriff auf alle Hardware- und Software-Features der Plattform nutzen können, ermöglicht extrem niederschwelligen Zugang zu Trusted-Computing-Konzepten. Mittlerweile wurde auch Apples iOS um ähnliche Konzepte erweitert [3] .

2. *Breitenverfügbarkeit* : Android ist klarer Marktführer im Bereich mobiler Plattformen mit über 70% Marktanteil [8]. Dadurch kann von einer globalen, breiten Verfügbarkeit der diskutierten Trusted-Computing-Features ausgegangen werden.

3 https://developer.apple.com/news/?id=2sngpulc 3. *Flexibilität und konzeptionelle Einfachheit* : Auf technischer Ebene wird ein Vertrauensverhältnis zu einem Android-Gerät durch die Verifikation einer X.509-Zertifikatskette hergestellt. Infolgedessen ergeben sich niederschwellige Integrationsmöglichkeiten in bestehende Systeme. Eine Konsequenz aus Punkt 1 ist Flexibilität und konzeptionelle Einfachheit auch insofern, als dass Entwicklerinnen und Entwicklern keine Einschränkungen in Bezug auf Programmkomplexität, oder durch Fehlen von trusted IO auferlegt werden. Dadurch lassen sich prinzipiell beliebig komplexe Anwendungen umsetzen, welche auf Benutzereingaben und –daten, sowie auf Sensoren und andere vorhandene Eingabegeräte zurückgreifen können. Im Kontext von Intels Software Guard Extensions (SGX), welche Trusted Computing am Desktop ermöglichen, ist dies hingegen nicht möglich.

4. *Offline Verifikation* : Der Vertrauensstatus eines Android-Geräts kann offline, d.h. ohne dass Dritte kontaktiert werden müssen, ermittelt werden. Dadurch ergibt sich die Möglichkeit, diese Umsetzung von Trusted Computing auch in dezentralen Szenarien einzusetzen.

Die Kombination dieser Eigenschaften beschreibt Trusted-Computing-Gesichtspunkte, welche aktuell nur durch aktuelle Android-Geräte erfüllt werden. Daraus ergibt sich insbesondere im Kontext dezentraler Systeme enormes Potential, wobei dies prinzipiell auf jede Plattform zutrifft, welche diese Eigenschaften aufweist. Entsprechend lassen sie diese Charakteristika als Anforderungen an generische Trusted-Computing-Systeme, welche in dezentralen Umgebungen eingesetzt werden sollen, formulieren. Tatsächlich ist es jedoch bisher auf globaler Ebene nur unter Android möglich beliebige sensible Operationen auf entfernte Geräte auszulagern, ohne dass zuvor eine Enrolment-Prozedur stattgefunden haben muss.

3. Vertrauensmodelle

Vertrauensmodelle können auf unterschiedlichste Arten und in diversen Detailgraden klassifiziert werden. Eines der charakteristischsten Merkmale eines Vertrauensmodells ist jedoch das Vorhandensein (oder Nichtvorhandensein) globaler Vertrauensanker auf deren Basis eine Vertrauenshierarchie umgesetzt werden kann. Sind solche Vertrauensanker vorhanden, ergeben sich klare Rollen innerhalb einer Hierarchie, was wiederum eine konzeptionell einfache, automatisierte Auswertung von Vertrauensketten ermöglicht. Im Gegensatz dazu lassen sich ohne derart klar definierte Rollen im Falle organisch wachsender Vertrauensstrukturen zwar von allen Teilnehmerinnen und Teilnehmern sehr individuelle Akzente in Bezug auf höchstpersönliche Vertrauenswahrnehmungen setzen, allerdings führt dies zu eher chaotischen Vertrauensgraphen, welche im Allgemeinen nicht automatisiert ausgewertet werden können. Buchmann, Karatsiolis und Wiesmaier [9] verwenden für diese konträren Vertrauensmodelle die Bezeichnungen *hierarchisches Vertrauen(smodell)* und *Web of Trust* . In hierarchischen Vertrauensmodellen wie PKIX wird (üblicherweise in Form eines X.509-Zertifikats) zertifizierten Eigenschaften einer Entität vertraut, wenn ein gültiger (Zertifikats)pfad zum Vertrauensanker gebildet werden kann. Die Organisation von Hierarchien und Vertrauensankern kann jedoch je nach Domäne stark variieren, wie im nachfolgenden Abschnitt erläutert.

3.1. Hierarchische Vertrauensmodelle

Einerseits kommen Teilnehmerinnen und Teilnehmer innerhalb hierarchischer Vertrauensmodelle über gemeinsame Vertrauensanker überein, gleichzeitig sind dadurch nicht automatisch alle Teilnehmerinnen und Teilnehmer vertrauenswürdig. Betrachtet man beispielsweise PKIX im Kontext von HTTPS, wird schnell ersichtlich, dass eine verhältnismäßig kleine Untermenge von Entitäten (Zertifizierungsstellen und

Webservices, welche über HTTPS verfügbar sind) einer großen Mehrheit prinzipiell nicht vertrauenswürdiger Instanzen gegenübersteht (von Nutzerinnen und Nutzern verwendete Webbrowser). Zwar werden clientseitige Zertifikate von allen namhaften Webbrowsern unterstützt, allerdings bietet nur eine verschwindend geringe Menge an Diensten zertifikatsbasierte Authentifizierung an. Stattdessen werden üblicherweise wissensbasierte Methoden, wie die Eingabe von Benutzername und Passwort herangezogen. Generell gibt es jedoch Alternativen zur Organisation dieses verbreiteten Vertrauensmodells, welche nachfolgend beleuchtet werden.

Basis für eine solche Gegenüberstellung unterschiedlicher Organisationsstrukturen hierarchischer Vertrauensmodelle bildet ein 1999 veröffentlichter Artikel von Radia Perlman [2]. Diese Quelle wurde aus einem wesentlichen Grund als Ausgangspunkt für den Vergleich und die Herausarbeitung von Stärken und Schwächen verschiedener Modelle gewählt: Durch das Alter des Artikels gibt dieser einen Blick auf mögliche Formalisierungsformen von Vertrauen im PKI-Kontext frei, welcher nicht vom mittlerweile seit Jahrzehnten aufrechterhaltenen Status quo im Bereich PKIX-Hierarchien beeinflusst ist. In diesem Kontext sind insbesondere zwei Ausprägungen von Vertrauenshierarchien relevant, welche in den folgenden zwei Unterabschnitten charakterisiert werden.

3.1.1. Einsatz eines einzigen universellen Vertrauensankers

Der Einsatz eines einzigen weltweiten Vertrauensankers wirkt unrealistisch bis bizarr, ermöglicht aber extrem simples Vertrauensmanagement. In der Praxis würde dies bedeuten, dass jegliche Hardware und Software von ein und derselben PKI mit eben einem einzigen globalen Vertrauensanker (auch über administrative Domänen hinweg) abhängig wäre. Zertifikate müssten folglich von einer einzigen Zertifizierungsstelle (*certification authority* (CA)) beantragt werden. Die Problematik dieses seltsam anmutenden Szenarios wurde von Perlman aus folgenden Gründen für untauglich befinden:

- Es ist unwahrscheinlich, dass alle Individuen und Organisationen weltweit tatsächlich einer einzigen Zertifizierungsstelle vertrauen würden.
- Die Prozeduren, um sich ein Zertifikat ausstellen zu lassen dürften nur wenig praktikabel und potentiell unsicher realisierbar sein, wenn man bedenkt, dass Zertifikate (unabhängig von der eigenen geografischen Lage) immer bei derselben (potentiell weit entfernten) Stelle beantragt werden müssen. Konkret wird die Problematik der Identitätsverifikation von Antragstellerinnen und Antragstellern aufgeworfen.
- Sollte das (einzige, universelle) Stammzertifikat unerwartet widerrufen oder ausgetauscht werden, wäre mit massiven Serviceausfällen (potentiell globalen Ausmaßes) zu rechnen. [4]
- Die universelle Zertifizierungsstelle könnte ihre Marktmacht missbrauchen.

Perlman diskutiert auch eine Abwandlung dieses Modells durch den Einsatz von sogenannten Registrierungsstellen (im englischsprachigen Originalartikel als *registration authorities* (RAs) bezeichnet). Diese würden im Zuge von Zertifikatsanträgen die Identitäten der Antragstellerinnen und Antragsteller überprüfen. Geprüfte Anträge würden von RAs entsprechend signiert an die universelle Zertifizierungsstelle weitergeleitet, welche anschließend die Zertifikate ausstellen würde. Auch wenn dieser Ansatz nach wie vor realitätsfremd wirken mag, besticht er unbestreitbar durch seine konzeptionelle Einfachheit und bietet auch unter realistischen Gesichtspunkten Vorteile, die das aktuelle PKIX-System nicht im Stande ist zu erbringen

(siehe nachfolgender Abschnitt): Weder müssen Stammzertifikate elaboriert verwaltet werden (da es nur eines gibt), noch stößt das Beantragen von Zertifikaten auf praktische Probleme. Signaturschlüssel der Registrierungsstellen können im Falle von Kompromittierungen einfach widerrufen werden und RAs können auch in geografischer oder organisatorischer Näher zu Antragstellern betrieben werden. Nichtsdestotrotz verbleiben klare Probleme, welche diesen Ansatz insgesamt unpraktikabel machen.

3.1.2. Einsatz mehrerer globaler Vertrauensanker

Im Rahmen dieses Modells kommen mehrere Zertifizierungsstellen (und entsprechend auch mehrere Stammzertifikate) zum Einsatz. In einfachsten Fall erfolgt die Zertifikatsausstellung analog zum vorigen Modell und Antragstellerinnen und Antragsteller suchen eine Zertifizierungsstelle ihrer Wahl auf. Laut Perlman ergeben sich dadurch im Vergleich zum Modell mit einer universellen Vertrauensstelle gleich auf zwei Ebenen entscheidende Sicherheitseinbußen: Einerseits genügt es, 4 Selbst im Falle eines kontrollierten Zertifikatsrollover wäre realistisch mit Problemen zu rechnen.

bereits eine Zertifizierungsstelle zu kompromittieren, um die Sicherheit des gesamten Vertrauensmodells zu unterwandern, andererseits ist es im Rahmen dieses Modells für Angreiferinnen und Angreifern verhältnismäßig leicht möglich, nichtsahnenden Nutzerinnen und Nutzern ein illegitimes Stammzertifikat unterzujubeln – schließlich fällt eine derartige Erweiterung beim Vorhandensein einer potentiell langen Liste an Vertrauensankern unter realistischen Bedingungen nicht auf. In Anlehnung an die Einführung von Registrierungsstellen im zuvor diskutierten Vertrauensmodell kann der in diesem Unterabschnitt beschriebene Ansatz um Zwischen-CAs (*intermediate CAs*) erweitert werden. Dies entspricht dem PKIX-Vertrauensmodell, welches aktuell eingesetzt wird, verschärft jedoch die zuvor angeführten Sicherheitsbedenken weiter. Vorfälle wie die auch medial zigfach aufgearbeitete Kompromittierung der niederländischen *DigiNotar* -Zertifizierungsstelle [10] haben gezeigt, dass diese Bedenken durchaus nicht unbegründet sind.

Zwar werden im 1999 von Radia Perlman veröffentlichten Artikel weitere Vertrauensmodelle skizziert und diskutiert, diese haben jedoch geringe bis keine praktische Relevanz. Aus diesem Grund werden nachfolgend die Eigenschaften des Web-of-Trust-Modells erläutert, welches zumindest in Nischenbereichen in der Praxis eingesetzt wird.

3.2. Web of Trust

Der Web-of-Trust-Ansatz kann als Gegenpol zu hierarchischen Vertrauesnmodellen betrachtet werden, da bewusst Abhängigkeiten zu zentralisierten Strukturen (wie Zertifizierungsstellen) vermieden werden. Infolgedessen erscheint ein Web of Trust in dezentralen Szenarien attraktiv. Die bekannteste Inkarnation eines solchen Web of Trust ist *Pretty Good Privacy* (PGP) und wurde ebenfalls von Radia Perlman diskutiert und hierarchischen Modellen gegenübergestellt. Die damals dargelegt Beschreibung dieses Systems und die daraus gefolgerten Schlüsse haben seither nicht an Aktualität eingebüßt:

[E]ACH USER STARTS OFF BY CONFIGURING PUBLIC KEYS THAT THEY HAVE LEARNED OUT OF BAND. THEN THEY OBTAIN CERTIFICATES FROM PUBLIC DATABASES.[...] IT IS COMMON WHENEVER THERE ARE LARGE GATHERINGS OF COMPUTER-ORIENTED PEOPLE TO HAVE PGP KEY SIGNING PARTIES WITH ELABORATE RITUALS [...]. IF YOU KNOW THE PERSON, YOU SIGN A CERTIFICATE FOR THEM. THIS APPROACH DOES NOT SCALE BEYOND A RELATIVELY SMALL COMMUNITY OF TRUSTED INDIVI- DUALS. IMAGINE IF IT WERE THE PKI OF CHOICE FOR THE

INTERNET. HOW BIG WOULD THE DATA- BASE OF CERTIFICATES HAVE TO BE? […] THIS IS ALREADY UNWORKABLE, BUT THERE'S A SECOND FATAL PROBLEM. EVEN IF BY SOME MIRACLE YOU WERE ABLE TO FIND A PATH THAT MATHEMATI- CALLY CREATED A CHAIN FROM THE KEY YOU TRUST A PRIORI TO THE TARGET NAME, HOW WOULD YOU KNOW WHETHER YOU COULD TRUST THAT CHAIN? [… A] CERTIFICATE ONLY STATES THAT THE SIGNER VERIFIES THE IDENTITY OF THE SUBJECT. TRUST IS CONSIDERED A LOCAL MATTER. […] THE TRUST INFORMATION […] ONLY APPLIES TO THE FIRST LINK IN THE CHAIN. [5] *[2, p. 40]* Im Artikel von 1999 wurde dieses Vertrauensmodell schlicht als *Anarchie* bezeichnet. Auch wenn dies eine pointierte Formulierung ist, fußt sie auf wohlbegründeten Argumenten auf, welche im Prinzip auf alle Vertrauensmodelle ohne globale Vertrauensanker anwendbar sind. Eines dieser Argumente ist die Tatsache, dass es gute Gründe dafür gibt, Vertrauen als eine lokale, persönliche Angelegenheit zu betrachten. Beispielsweise ist es durchaus schlüssig, die Identität einer an sich 5 *Jeder Benutzer konfiguriert zu Beginn public Keys, welche abseits der üblichen Kommunikationskanäle übermittelt wurden. Anschließend werden Zertifikate aus öffentlichen Datenbanken abgerufen. […] Wann immer größere Mengen computerorientierter Menschen zusammenkommen, ist es üblich, PGP-Key-Signing- Parties mit elaborierten Ritualen zu veranstalten […]. Kennt man eine Person, signiert man deren Zertifikat. Dieser Ansatz skaliert nicht über eine relativ kleine Gemeinschaft vertrauenswürdiger Individuen hinaus. Man stelle sich vor, dass dies die PKI der Wahl für das Internet wäre. Wie groß müsste die Zertifikatsdatenbank sein? […] Bereits das ist nicht machbar, und es gibt noch ein zweites Problem. Selbst wenn man es wie durch ein Wunder schaffen würde, eine Kette von einem a-prior vertrauenswürdigen Schlüssel zum gesuchten Ziel zu erstellen, woher sollte man wissen, dass diese Kette vertrauenswürdig ist? [… Ein] Zertifikat bezeugt lediglich, dass der Signierende die Identität eines anderen bestätigt. […] Vertrauen wird als eine lokale Angelegenheit betrachtet. […] Die Vertrauensinformation […] bezieht sich nur auf das erste Glied in der Kette.* nicht vertrauenswürdigen Person zu zertifizieren. Die Problematik liegt vielmehr in der Konsequenz dieses Ansatzes: Das Vertrauen in auf diese Weise zustande gekommene Vertrauensketten ist inhärent zweifelhaft, da keine Aussage über die Vertrauenswürdigkeit der signierenden Parteien getroffen werden kann. Zwar definieren aktuelle *OpenPGP* [6] -konforme PGP-Implementierungen wie *GnuPG [7]* unterschiedliche Vertrauensstufen (*Level of Trust*), um Vertrauensverhältnisse akkurater abbilden zu können, allerdings ändert das nichts am zugrunde liegenden Problem. Darüber hinaus ist es ebenfalls nicht hilfreich, dass GnuPG das Vertrauensmaß, das ein Benutzer oder eine Benutzerin einem Schlüssel entgegenbringt, als private Information ansieht: „Das Vertrauensmaß eines Schlüssels ist etwas, das Sie alleine dem Schlüssel zuordnen, und es wird als private Information betrachtet. Es wird nicht mit dem Schlüssel verpackt, wenn dieser exportiert wird" [11, p. 32]. Da PGP oft im Umfeld von Graswurzelbewegungen eingesetzt wurde (und wird), kommt noch ein weiterer persönlicher Aspekt hinzu: Jede und jeder kann völlig frei und ungezwungen darüber entscheiden, was im Rahmen der eigenen persönlichen Einschätzung einem bestimmten Vertrauensmaß und bestimmten Vertrauensstufen entspricht. In letzter Konsequenz hat dieser Umstand zur Folge, dass jegliche automatisierten Versuche, Vertrauen entlang von Vertrauensketten zu propagieren und auszuwerten von vorn herein zum Scheitern verurteilt sind.

Ein Web of Trust, das wie PGP auf menschlichen Entscheidungen und höchstpersönlichen Bewertungen, und der Geheimhaltung von Vertrauenseinschätzungen basiert, wird folglich nie auf globaler Ebene einsetzbar sein, und lediglich innerhalb bestimmter Nischen erfolgreich sein. Ein solches Einsatzgebiert ist die Signatur von Softwarepaketen im Rahmen von Linuxdistributionen oder Quellcode. Zusammenfassend

lässt sich festhalten, dass ein Web-of-Trust-Ansatz ohne globale Vertrauensanker nur schwerlich im globalen Maßstab umsetzbar ist, da Vertrauensketten möglichst kurz gehalten werden müssen, um deren Aussagekraft zu bewahren.

4. Enrolment-lose Vertrauensbildung in dezentralen Umgebungen

Die in Abschnitt 2.3 aufgeführten Eigenschaften treffen prinzipbedingt auch auf die Arbeit an einem dezentralen, Trusted-Computing-basierten Peer-to-Peer-Netzwerk zu, welche im Rahmen eines vorangegangenen A-SIT-Projekts präsentiert wurden [3]. Betrachtet man diese Ergebnisse im Kontext der im Rahmen dieses Berichts dargelegten Diskussion zu Vertrauensmodellen, lassen sich eine Reihe von Eigenschaften bezüglich des eingesetzten Vertrauensmodells im dezentralen Peerto-Peer-Kontext ausmachen. Grundsätzlich setzt der Einsatz von Android als Trusted-Computing-Basis vertrauen in ein von Google ausgestelltes Stammzertifikat voraus. Unter diesem Gesichtspunkt handelt sich im Wesentlichen um ein Vertrauensmodell, welches sich auf einen universellen Vertrauensanker stützt. Insgesamt lässt dies folgende Schlüsse zu:

1. Der Einsatz eines universellen Vertrauensankers im globalen Maßstab ist innerhalb bestimmter Domänen möglich und praktikabel.

2. Das Ausrollen von Zertifikaten ist in der Praxis unproblematisch, da das im Zuge der Attestierung notwendige Schlüsselmaterial bei der Fertigung in Smartphones eingebracht wird.

3. Zertifikatswiderruf und die Auswertung von Widerrufsinformationen ist in der Praxis umsetzbar.

Während die ersten beiden Aspekte durch die Ergebnisse vorangegangenen A-SIT-Projekte untermauert werden, wurde auf die Verbreitung und Verarbeitung von Widerrufsinformationen bisher nicht eingegangen. Dabei handelt es sich unter realistischen Bedingungen jedoch um kritische Informationen, welche akkurat und zeitnah allen Teilnehmerinnen und Teilnehmern eines Vertrauensmodells verfügbar gemacht werden müssen. Wie erwähnt, stellt Google Widerrufsinformationen zu den in Android-Geräten eingebrachten Zertifikaten zur Verfügung, allerdings muss hierfür ein Google-Service kontaktiert werden. Im Kontext einer Offline-Verifikation, 6 https://www.openpgp.org/ 7 https://www.gnupg.org/ wie sie eingangs beschrieben wurde, sowie im Rahmen dezentraler Operation eines verteilten Systems, stellt dies auf den ersten Blick einen nicht überwindbaren Widerspruch dar. Auf Grund der Tragweite dieses Umstandes wir diese Thematik im nachfolgenden Abschnitt detailliert diskutiert, sowie eine konkrete Lösung für dieses augenscheinlich unlösbare Problem vorgestellt.

4.1. Dezentrale Verteilung von Widerrufsinformationen

Die Art und Weise, wie Widerrufsinformationen im Kontext von Remote Attestation unter Android zur Verfügung gestellt werden, ist nicht mit den von der *Internet Engineering Task Force* veröffentlichten X.509-Spezifikationen kompatibel. Die dem Konzept zu Grunde liegende Semantik ist jedoch äquivalent zu den im Rahmen von X.509 standardisierten *Certificate Revocation Lists* (CRLs) [12]: Ein von Google betriebener Webservice stellt ähnlich wie ein traditioneller *CRL Distribution Point* eine statische Liste zu Verfügung, welche die Identifikatoren aller widerrufenen Signaturschlüssel, bzw. Zertifikate enthält. Diese Datenstruktur beinhaltet jedoch weder Informationen bezüglich der Aktualität der bereitgestellten Informationen, noch ist

diese kryptografisch signiert, oder beinhaltet sonstige Hinweise auf Mechanismen zu Authentizitäts- und Integritätsprüfung. Tatsächlich werden jedoch all diese kritischen Informationen auf anderem Wege vermittelt:

Authentizität: Herkunftsauthentizität wird mittels *Transport Layer Security* (TLS) sichergestellt: Ein im Browser, bzw. Betriebssystem vorkonfigurierter Vertrauensanker ermöglicht es, eine Zertifikatskette bis zum vom Webservice für die Auslieferung der Widerrufsinformationen verwendeten Zertifikat aufzubauen. Durch den Einsatz von TLS sind alle übertragenen Daten signiert und deren Herkunft kann nachgewiesen werden.

Integrität: Die Integrität aller übertragenen Daten wird durch dein Einsatz von Authenticated Encryption im Rahmen einer TLS 1.3 Cipher-Suite sichergestellt.

Aktualität: Die Widerrufsinformationen werden innerhalb eines TLS-Datenstroms per HTTP übertragen. Die Serverantwort auf eine Anfrage nach der Widerrufsliste beinhaltet alle nötigen Aktualitätsinformationen im HTTP-Nachrichtenkopf. Tabelle 1 veranschaulicht diesen Umstand an Hand eines aufgezeichneten HTTP-Nachrichtenkopfs.

Zwar lässt sich hiermit festhalten, dass alle nötigen Informationen um Authentizität, Integrität und Aktualität der Widerrufsliste zu verifizieren vorhanden sind, jedoch ergeben sich diese aus einem innerhalb einer TLS-Sitzung gekapselten HTTP-Datenstroms. Da dieser Datenaustausch mit einem von Google betriebenen Webservice stattfinden, handelt es sich dabei augenscheinlich nicht um Offline-Verifizierbarkeit von Widerrufsinformationen – und erst recht nicht in einem dezentralen Kontext. Um dies dennoch zu erreichen, muss ein Weg gefunden werden, welcher es ermöglicht, diese kritischen Daten, welche im HTTP-TLS-Datenstrom enthalten sind, unabhängig von einer TLS-Sitzung verfügbar zu machen.

Tatsächlich lässt sich eine TLS-Sitzung verhältnismäßig einfach serialisieren und folglich auch replizieren. Im Prinzip wäre es somit möglich, einmal abgerufene Widerrufsinformationen samt HTTP-Nachrichtenkopf (welcher Aktualitätsinformationen enthält) innerhalb einer aufgezeichneten und serialisierten TLS-Sitzung (welche Integrität und Authentizität der Daten garantiert) beliebig zu vervielfältigen und zu verteilen, ohne dass der Webservice innerhalb der Gültigkeitsdauer des Datensatzes kontaktiert werden müsste. Folglich wäre es analog zu *OCSP Stapling* [13] möglich, Widerrufsinformationen im Kontext eines Attestierungsvorgangs mitzuübertragen, wodurch tatsächlich eine Offline-Verifikation, bzw. Offline-Attestierung möglich gemacht würde. Hierbei gilt es zu beachten, dass es sich um *globale* Widerrufsinformationen handelt. D.h. die angefragte Widerrufsliste deckt das gesamte relevante Schlüsselmaterial des gesamten Android-Ökosystems im Kontext von Remote Attestation ab. Daher wäre es möglich, die Liste einmalig innerhalb eines 24h-Intervalls abzurufen und unter allen Geräten (beispielsweise über ein Peer-to-Peer-Netzwerk) zu verteilen. Damit würde zwar der periodische Abruf von Widerrufsinformationen nach wie vor in Abhängigkeit zu einer zentralen Instanz stehen, allerdings wird eine Offline-Verifikation im Regelbetrieb möglich.

Headername Wert Accept-Ranges bytes **Vary** Accept-Encoding **Content-Type** application/json **Cross-Origin-Resource-Policy** cross-origin **Content-Length** 1524

Cache-Control public, max-age=86400 **Last-Modified** Tue, 17 Nov 2020 18:45:00 GMT **X-Content-Type-Options** nosniff **Server** sffe **X-XSS-Protection** 0 **Alt-Svc** h3-29=":443"; ma=2592000,h3-...

Tabelle 1: HTTP-Nacrichtenkopf der Serverantwort, welche die Widerrufsliste beinhaltet. Informationen, welche die Aktualität der ausgelieferten Daten bescheinigen, sind grau hinterlegt. Um dieses Loslösen vom Widerrufsservice tatsächlich umsetzen zu können, muss jedoch ein weiteres Problem gelöst werden: Die im Rahmen von TLS 1.3 eingesetzten Cipher-Suites erzwingen Forward Secrecy, verwenden also nach erfolgter wechselseitiger Authentifizierung Wegwerf-Schlüssel. Daher ist es nicht ohne weiteres möglich, aufgezeichnete TLS-Sitzungen zu entschlüsseln, da zusätzlich zum Server-Zertifikat noch die Clientseitigen Wegwerfschlüssel benötigt werden, um die im Rahmen des TLS-Handshake-Protokolls ausgehandelten zur Verschlüsselung des Datenstroms verwendeten Schlüssel wiederherstellen zu können. Zwar handelt es sich dabei prinzipiell um sensible Informationen, da jedoch *innerhalb* des TLS-Datenstroms keine geheimen Nachrichten übertragen werden, können diese mitsamt einer aufgezeichneten TLS-Sitzung weitergegeben werden. Die Kombination von aufgezeichneter TLS-Sitzung, serverseitigem Zertifikat und clientseitigen Wegwerfschlüsseln erlaubt es folglich die ursprüngliche TLS-Sitzung inklusive HTTP-Datenstrom (und somit auch der im HTTP-Nachrichtenkopf enthaltenen Aktualitätsinformationen) reproduzierbar weiterzugeben. Implementierungstechnisch gibt es in diesem Zusammenhang keine Hindernisse, da im Rahmen des *Wireshark* -Projekts [8] bereits entsprechende Programmbibliotheken erstellt wurden, welche alle benötigten Funktionalitäten zur Verfügung stellen.

4.2. Ein praktikables, hybrides Vertrauensmodell

Ausgehend von den im Rahmen dieses Berichts bisher diskutierten Aspekten lässt sich eine Definition für ein hybrides, praktisch einsetzbares Vertrauensmodell ableiten, welches die Flexibilität eigens Web of Trust mit der automatisierten Verifizierbarkeit hierarchischer Vertrauensketten kombiniert. Zwar muss dafür einer zentralen, universellen Zertifizierungsstelle (Google) vertraut werden, allerdings ergeben sich dadurch bisher nur theoretisch abgehandelte Möglichkeiten, die auch praktisch umsetzbar sind. Wie in Abschnitt 3.2 erwähnt, müssen Vertrauensketten im Kontext eines Web of Trust kurz gehalten werden, um ihre Aussagekraft zu behalten. Der Einsatz von Remote Attestation, um Android-Geräte als Trusted-Computing-Basis einsetzen zu können, ermöglicht dies: Einerseits kann auf Basis eines Attestierungsergebnisses ein individuelles Vertrauensniveau ausgesprochen werden (siehe Abschnitt 2.2), andererseits, bildet dieser individuelle Vertrauensausdruck lediglich das letzte Glied einer Vertrauenskette, welche im Kern über eine PKIX-Hierarchie aufgebaut wird. Die Eigenschaften dieses hybriden Vertrauensmodells lassen sich wie folgt zusammenfassen:

8 https://www.wireshark.org/

- Der Einsatz eines universellen Vertrauensankers erlaubt es mittels Remote Attestation, aktuelle Android-Smartphones in den Status einer Trusted-Computing-Basis zu erheben.
- Es ist nicht notwendig, Dritte in den Prozess der Attestierung, bzw. der Vertrauensbildung miteinzubeziehen. Dies ist im Kontext einer Offline-Vertrauensbildung, bzw. in dezentralen Szenarien relevant.
- Vertrauen ist kein binärer Wert, sondern kann im Rahmen persönlichen Ermessens granular formuliert werden. Beispielsweise kann miteinbezogen werden, ob ein Gerät, dessen Vertrauensmaß bestimmt werden soll, am aktuellen Sicherheitsupdate-Stand ist, oder noch eine veraltete Betriebssystemversion verwendet.

- Es gibt keine vordefinierten Richtlinien oder Normen, welche vorgeben, an welche Eigenschaften ein bestimmtes Vertrauensmaß gekoppelt sein muss. Letztendlich handelt es sich dabei um eine persönliche Entscheidung.
- Ist Vertrauen zu einem Gerät im Sinne einer Trusted-Computing-Basis hergestellt, gibt es keinerlei Limitierungen im Vergleich zur Ausführung von Operationen auf nicht vertrauenswürdigen Geräten.

In traditionellen Web-of-Trust-Modellen stehen Individuen im Vordergrund, während im Kontext dieses Projekts Geräten Vertrauen ausgesprochen wird. In der Praxis hat dies jedoch insofern keine Auswirkung, als dass Geräte im Sinne des Vertrauensmodells als Container für Schlüsselmaterial dienen. Unter diesem Gesichtspunkt ist dies äquivalent zur Verwendung von kryptografischen Hardware-Token, welche auch mit PGP eingesetzt werden können. Daher ist es letztendlich irrelevant, ob direkt Geräten, welche durch kryptografisches Material identifiziert werden, Vertrauen ausgesprochen wird, oder einer Person, welche auf technischer Ebene ebenfalls durch kryptografisches Material repräsentiert wird.

5. Conclusio

Im Rahmen dieses Projekts wurden Forschungsstränge, welche Android als Trusted-Computing-Basis nutzbar machen dahingehend abgerundet, als dass ein hybrides Vertrauensmodell definiert wurde, welches in diesem Kontext eingesetzt werden kann. Im Zuge dieser Arbeit wurden hierarchische und dezentrale Vertrauensmodelle charakterisiert, bzw. auf eine mehr als zwanzig Jahre zurückliegende Analyse dieses Themenkomplexes zurückgegriffen, deren Schlussfolgerungen nicht durch die Omnipräsenz und den dogmatischen Zugang zu PKIX beeinflusst waren. In diesem Kontext wurden die Vor- und Nachteile unterschiedlicher Vertrauensmodelle beleuchtet. Aufbauend auf den daraus resultierenden Erkenntnissen wurde durch den Einsatz von Trusted Computing ein Vertrauensmodell geschaffen, welches erstmalig eine Umsetzungsmöglichkeit bietet, Vertrauen in Dritte auf persönlicher Ebene zu definieren, ohne dabei auf praktische Probleme, wie im Kontext von PGP zu stoßen. Jede Teilnehmerin und jeder Teilnehmer dieses hybriden Vertrauensmodells entscheidet selbst welches Maß an Vertrauen sie oder er Dritten entgegenbringt. Da jedoch zu keinem Zeitpunkt Vertrauensketten aufgebaut werden, deren Aussagekraft sich auf diese persönlichen Urteile anderer stützen, weist die im Rahmen dieses Projekts erarbeitete Lösung keine der Schwächen von Web-of-Trust-Modellen auf. Lediglich das letzte Glied einer Vertrauenskette, welches nur im Kontext einer direkten Kommunikation mit der oder dem Betroffenen relevant ist, reflektiert das persönliche Vertrauensmaß, welches der oder demjenigen gegenüber ausgesprochen wird. Um den dezentralen Charakter eines Web of Trust auch im Betrieb aufrechtzuerhalten, wurde überdies ein Konzept erarbeitet, welches die von einer zentralen Instanz losgelöste Verbreitung von Widerrufsinformationen ermöglicht.

Referenzen

[1] E. Rescorla, „The Transport Layer Security (TLS) Protocol Version 1.3," Internet Engineering Task Force, 08 2018. [Online]. Available: https://tools.ietf.org/html/rfc8446. [Zugriff am 26 03 2021].

[2] R. Perlman, „An overview of PKI trust models," *IEEE Network,* pp. 38-43, 11 1999.

[3] B. Prünster, E. Fasllija und D. Mocher, „Master of Puppets: Trusting Silicon in the Fight for Practical Security in Fully Decentralised Peer-to-Peer Networks," in *Proceedings of the 16th International Security and Cryptography (SECRYPT)*, SciTePress - Science and Technology Publications, 2019.

[4] B. Prünster, G. Palfinger und C. Kollmann, „Fides - Unleashing the Full Potential of Remote Attestation,“ in *Proceedings of the 16th International Security and Cryptography (SECRYPT)*, Prag, Tschechien, SciTePress - Science and Technology Publications,, 2019.

[5] B. Prünster, „Bereitstellung von Smartphone-Features am Desktop,“ A-SIT, 10 10 2020.

[Online]. Available: https://technology.a-sit.at/veroeffentlichungen/bereitstellung-von-smartphone-features-am-desktop/. [Zugriff am 26 03 2021].

[6] R. Mayrhofer, J. Vander Stoep, C. Brubaker und N. Kralevich, „The Android Platform Security Model,“ 11 04 2019. [Online]. Available: http://arxiv.org/abs/1904.05572. [Zugriff am 11 06 2019].

[7] B. Prünster, „Erhebung State-of-the-Art Remote Attestation,“ A-SIT, 17 06 2019. [Online].

Available: https://technology.a-sit.at/veroeffentlichungen/erhebung-state-of-the-art-remote-attestation/. [Zugriff am 01 09 2020].

[8] StatCounter, „Mobile Operating System Market Share Worldwide Jan 2020 - Jan 2021,“ 02 2021. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide.

[Zugriff am 26 03 2021].

[9] J. A. Buchmann, E. Karatsiolis und A. Wiesemaier, Introduction to Public Key Infrastructures, Berlin, Heidelberg: Springer, 2013.

[10] European Network and Information Security Agency, „Operation Black Tulip: Certificate authorities lose authority,“ 2011. [Online]. Available: https://www.enisa.europa.eu/media/news-items/operation-black-tulip. [Zugriff am 26 03 2021].

[11] Free Software Foundation, Inc., „Das GNU-Handbuch zum Schutze der Privatsphäre,“ 2000.

[Online]. Available: https://www.gnupg.org/gph/de/manual.pdf. [Zugriff am 26 03 2021].

[12] R. Housley, W. Ford, W. Polk und D. Solo, „Internet X.509 Public Key Infrastructure Certificate and CRL Profile,“ 01 1999. [Online]. Available: http://www.rfceditor.org/rfc/rfc2459.txt. [Zugriff am 26 03 2012].

[13] D. E. 3rd, „ Transport Layer Security (TLS) Extensions: Extension Definitions,“ 01 2011.

[Online]. Available: https://www.rfc-editor.org/rfc/rfc6066.txt. [Zugriff am 26 03 2021].