16. Dezember 2016 Studie · Digitale Identität · Transparenz · 21 Min. Lesezeit

CA-lose Authentifizierung von Clouddiensten

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

Quelle https://technology2.a-sit.at/veroeffentlichungen/ca-lose-authentifizierung-von-clouddiensten/

Dieses Projekt evaluiert verschiedene Verfahren zur CA-losen Authentifizierung von Entitäten und Services.

Durch den Vormarsch moderner Webtechnologien wie WebRTC ist es immer einfacher möglich, vor allem kurzlebige Dienste von Endgeräten aus anzubieten. Genau wie im Fall statischer, bzw. stationärer Dienste stellt sich auch hier die Frage nach der Authentifizierung angebotener Dienste. Im Rahmen dieses Projekts wurden unterschiedliche Verfahren evaluiert, die potentiell dafür geeignet sind, Entitäten und Services ohne den Einsatz von Zertifizierungsdiensten zu authentifizieren.

Dazu wurden einerseits Ansätze untersucht, die soziale Netze von Nutzerinnen und Nutzern heranziehen, andererseits aber auch Befehlssatz-Erweiterungen für CPUs näher betrachtet, welche hardwaregestützte Zugriffskontrollen anbieten und somit Trusted Execution auch auf an sich nicht vertrauenswürdigen Geräten ermöglichen. Zusätzlich wurden auch reine Software-Lösungen untersucht. Im Zuge dieses Projekts wurde außerdem ein Prototyp auf Basis des Inverted Trust Model umgesetzt, um die Praxistauglichkeit einiger der zuvor diskutierten Konzepte zu demonstrieren.

1. Einleitung

Moderne Applikationen bedienen sich vermehrt der Cloud, um zusätzlich benötigte Ressourcen zu allokieren. Dabei ergeben sich zweierlei Unzulänglichkeiten. Erstens gibt es vermehrt temporäre Cloud-Dienste, die nur sehr kurzlebig sind und darum nicht zwingend über eine statische und prüfbare Identität verfügen. Ein Beispiel dafür ist ein Browser, der eine vorübergehend freie Rechnerkapazität seines Hosts (eines Smartphones) einem geografisch benachbartem Smartphone zur Verfügung stellt. Es ist nicht wirtschaftlich, für solch kurzlebige Dienste je einen Zertifizierungsprozess durchzuführen bzw. das Ergebnis der Zertifizierung zu verteilen. Zweitens bedienen sich aktuelle Applikationen oft moderner Kommunikationstechnologien. Beispiele dafür sind HTML5 *Cross-Document Messaging* oder *WebRTC*. Diesen Technologien © A-SIT, Partner für Sicherheitsfragen steht kein *Domain Name System* (DNS) zur Verfügung, das einen klassischen Zertifizierungsprozess mittels Zertifizierungsdienst ermöglichen würde. Es gibt jedoch aus anderen Forschungsgebieten Lösungsansätze für die oben genannten Probleme. Beispiele sind die Verwendung von Social-Media-Services zum Feststellen von Schlüsselauthentizität, das

Aufbauen einer Trust Relationship auf einer impliziten Information, ohne die ein Cloudservice seine Aufgabe nicht erfüllen könnte, oder Ansätze auf Protokollebene wie *Off-the-Record Messaging* (OTR) und das *Secure Real-time Transport Protocol* (SRTP) bzw. ZRTP. Dieses Projekt diskutiert vielversprechende Methoden zur Authentifizierung und evaluiert deren Eignung für Anwendungsfälle in denen klassische Zertifizierung entweder nicht wirtschaftlich sinnvoll oder gar nicht möglich ist. Dabei wird zwischen Technologien unterschieden, die lediglich die Authentifizierung einer Entität zum Ziel haben ohne eine Identifizierung zu liefern und Technologien die Authentifizierung und Identifizierung anbieten. Zwei Prototypen demonstrieren, dass es sehr wohl möglich ist, Authentifizierung und auch Identifizierung in modernen Systemen, die nicht kompatibel zu klassischen Zertifizierungsdiensten und PKIs sind, umzusetzen. Es zeigt sich allerdings, dass die Voraussetzungen und Anforderungen sehr speziell sein müssen, um ausreichend sichere Lösungen zu ermöglichen. Sehr schnell stößt man auf die generelle Identifikationsproblematik, die bereits im nicht-technischen Umfeld nur näherungsweise lösbar scheint. Alles in allem halten neue Technologien schnell Einzug in alltägliche Anwendungsfälle, technische Lösungen zu Authentifizierung und Identifizierung und damit auch zu Datenschutz und Privatsphäre können allerdings nicht immer Schritt halten. Dieses Projekt zeigt auf, dass unter gewissen Voraussetzungen durchaus zufriedenstellende alternative Lösungen möglich sind, man jedoch sehr schnell an der generellen Identifikationsproblematik angekommen ist. Zunächst werden Authentifizierungstechniken diskutiert, die ohne Identifikator auskommen bzw. die keinen Identifikator benötigen. Nachfolgend werden Authentifizierungstechniken behandelt, die auch einen Identifikator liefern. Beide Abschnitte evaluieren die Eignung der jeweiligen Technologien für moderne Anwendungsfälle und Infrastrukturen. Abschließend werden die Prototypen im Hinblick auf die nötigen Voraussetzungen und die realisierbaren Anwendungsfälle diskutiert. Eine Zusammenfassung schließt den Bericht ab.

2. Authentifizierung ohne Identifikator

Bei der Authentifizierung ohne Identifikator geht es nicht um die Identifizierung bzw. Wiedererkennung von einzelnen Geräten, sondern darum, die Eignung eines Geräts für bestimmte Aufgaben festzustellen. Operationen auf bestimmten Daten und können auf andere, sich möglicherweise in der Umgebung befindliche Geräte ausgelagert werden. Dabei ist die Sensibilität der Daten entscheidend. Vorgänge, die auf kritischen Daten ausgeführt werden, dürfen nur auf vertrauenswürdige Geräte ausgelagert werden, wohingegen unkritische Operationen auf unkritischen Daten auch auf weniger vertrauenswürdige Geräte ausgelagert werden könnten. Dabei muss eine geeignete Abstraktionsschicht gefunden werden und zwischen verschiedenen Architekturen unterschieden werden:

```
- Mobile Geräte bieten, abhängig vom Betriebssystem, bereits eine hohe Isolation
zwischen Apps, bzw. von Benutzerinnen und Benutzern zu Applikationen und deren
Daten. Unter Einhaltung und Sicherstellung dieser Eigenschaft können Geräte dazu
verwendet werden, Tasks auszulagern.

- Für Desktop- oder Serversysteme wird eine andere Lösung benötigt. Hierbei muss
entweder dem Betreiber bzw. der Betreiberin der Infrastruktur vertraut werden, oder
es müssen Techniken wie _Intel Software Guard Extensions_ (SGX) [1] oder _Intel
Trusted Execution Technology_ (TXT) [1] verwendet werden. In den folgenden
Abschnitten werden Lösungsansätze für Mobil- und Desktopsysteme diskutiert.
```

1 http://www.intel.com/content/www/us/en/architecture-and-technology/trusted-execution-technology/trusted- execution-technology-security-paper.html Bei mobilen Geräten muss bedacht werden, dass die Vertrauensbewertung eines Geräts nur unter bestimmten Möglichkeiten akkurat durchgeführt werden kann. Ist ein Gerät gerootet, können keine Sicherheitsgarantien mehr abgegeben werden und eine Isolation zwischen Applikationen kann nicht mehr gewährleistet werden. Es gibt zwar Möglichkeiten, gerootete Geräte zu erkennen, diese arbeiten jedoch bei derzeitigen Android– bzw. iOS-Architekturen nicht zuverlässig.

2.1. Mobile Device Management

Nicht gerootete mobile Geräte bieten bereits eine hinreichende Abschirmung zwischen Applikationen. Unter der Voraussetzung, dass die Applikation die Sicherheitsmechanismen, die für mobile Applikationen zur Verfügung gestellt werden, in Anspruch nimmt, ist es Nutzern und Nutzerinnen nur bedingt möglich, auf Daten von Applikationen zuzugreifen. *Mobile Device Management* (MDM) wird von verschiedensten Herstellern umgesetzt wie *Android for Work* [2], *VMWare Airwatch* [3], oder *WSO2 Enterprise Mobility Manager* [4] zeigen. Diese bieten die Möglichkeit, Sicherheitseinstellungen am Gerät zu erzwingen und Aufgaben von entfernten Geräten aus auszuführen. MDM-Systeme sind für den Einsatz in Unternehmensumgebungen gedacht, um beispielsweise Sicherheitseinstellungen auf Geräten von Benutzerinnen und Benutzern in Bring-your-own-Device-Szenarien zu erzwingen. Um Geräte bzw. deren Vertrauenswürdigkeit sicherzustellen, können durch MDM-Lösungen beispielsweise automatisiert Zertifikate am Gerät installiert werden, die anschließend dafür verwendet werden können, Geräte mit aktivem MDM erkennen zu können. Das MDM erzwingt in diesem Fall Sicherheitseinstellungen und verifiziert, im Rahmen des Möglichen, die Sicherheit des Geräts.

2.2. Google SafetyNet

Bei *Google SafetyNet* [5] handelt es sich um einen ähnlichen Ansatz wie beim zuvor beschriebenen Mobile Device Management. Dabei wird ebenfalls darauf vertraut, dass auf nicht gerooteten Geräten, Daten von Applikationen so abgelegt werden können, dass keine anderen Entitäten darauf Zugriff erlangen können. Ähnlich wie bei MDM-Lösungen kann auf gerooteten Geräten die Datensicherheit nicht garantiert werden. SafetyNet ist ein Service, das auf allen von Google erweiterten mobilen Android-Geräten installiert ist und im Hintergrund läuft. Das SafetyNet Service im Hintergrund sendet Daten an SafetyNet-Server zur Analyse. Dieser Hintergrunddienst ist nicht Teil von Applikationen, verbindet sich zu Google-Servern und kann direkt Updates beziehen, ohne den Umweg über den *Play Store* zu nehmen. Es ist nicht bekannt, welche Daten genau für die Analyse herangezogen werden. Basierend auf diesen Daten wird ein Kompatibilitätswert errechnet, der angibt, ob es sich um ein bekanntes Gerät handelt, das in geeigneter Weise konfiguriert wurde. Eine daraus ableitbare Aussage ist, ob ein Gerät gerootet wurde oder sich im Originalzustand befindet. Dieses Ergebnis ist nur in gewissen Grenzen vertrauenswürdig, da gerootete Geräte prinzipiell vortäuschen können, nicht modifiziert zu sein. Am Ende kommt es darauf an, wie kritisch ein Datenverlust wäre und wo dieser im Risikomodell anzusiedeln wäre. Neben dem Hintergrunddienst stellt Google SafetyNet eine Attestation-API bereit, die es ermöglicht, eine Bewertung des aktuellen Geräts abzurufen. Das Attestation-Resultat ist als signierte JSON-Struktur abgebildet und beinhaltet folgende Punkte:

```
   - Einen Zufallswert, um die Aktualität des Ergebnisses sicherzustellen.

   - Den Paketnamen der Applikation (des APKs), die den Aufruf getätigt hat.

   - Kryptografische Hashes des Zertifikats das verwendet wurde, um die Applikation
zu signieren.
```

2 https://www.android.com/work/ 3 http://www.air-watch.com/solutions/android-management/ 4 http://wso2.com/products/enterprise-mobility-manager/ 5 https://developers.google.com/android/reference/com/google/android/gms/safetynet/SafetyNet

```
   - Angaben über den Zustand des Geräts, bzw. das Attestation-Ergebnis. Eine
mögliche Integration in ein Framework, bei dem von einem dritten Gerät (dem
_Requester_ ) ein Attestation-Ergebnis angefordert werden muss, ist in Abbildung
dargestellt:

   - Im ersten Schritt muss vom außenstehenden Benutzer bzw. von der außenstehenden
Benutzerin ein Zufallswert erzeugt werden, der die Aktualität des Attestation-
Ergebnisses bestätigt.

   - Der Zufallswert wird an das Zielgerät übermittelt, mit der Aufforderung eine
Attestation durchzuführen.

   - Das Zielgerät kontaktiert das SafetyNet-Webservice, und fordert mit dem
erhaltenen Zufallswert ein Attestation-Ergebnis an.

   - Das Attestation-Ergebnis wird zum außenstehenden Benutzer bzw. zur
außenstehenden Benutzerin weitergeleitet, welcher bzw. welche den Zufallswert
überprüfen kann.

   - Um die kryptografische Gültigkeit des Ergebnisses zu überprüfen, wird das
Ergebnis zur Verifikations-API weitergeleitet.
```

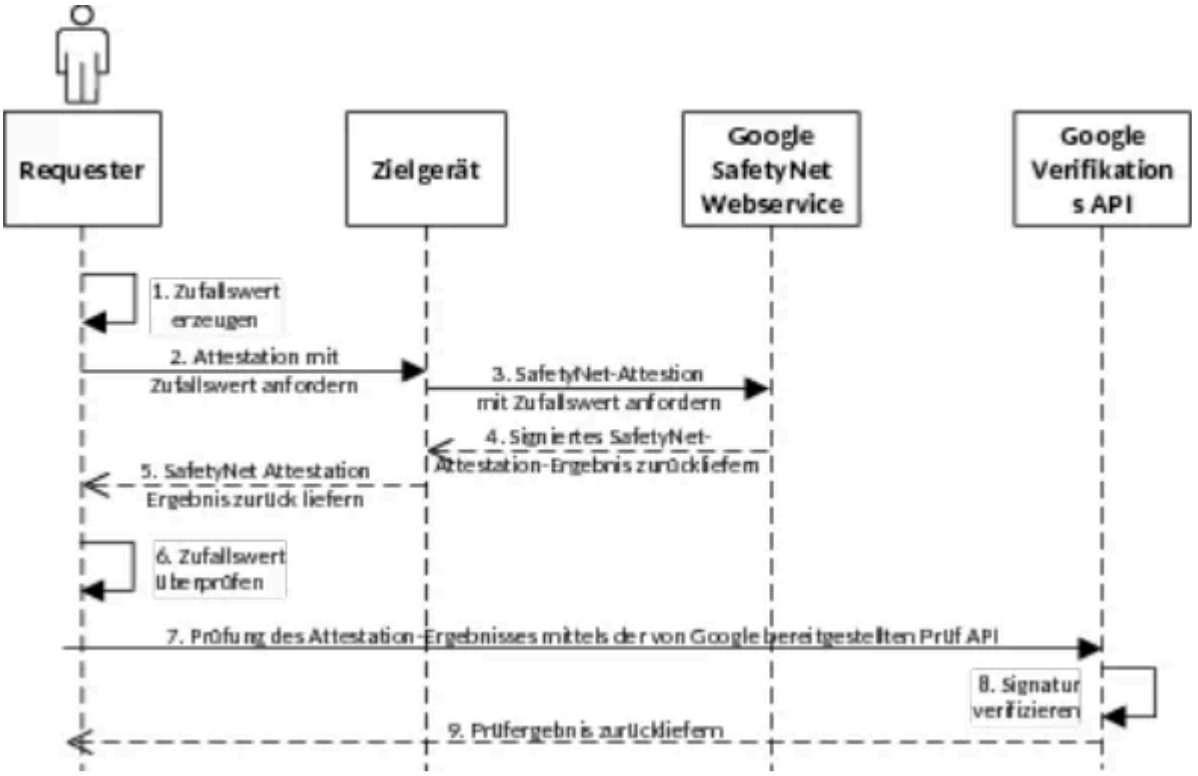
Abbildung 1: Google Safetynet Integration

```
   - Basierend auf dem Verifikationsergebnis kann die Benutzerin bzw. der Benutzer
entscheiden, ob das Gerät geeignet ist, eine Operation auf bestimmten Daten
auszuführen.
```

2.3. Intel Software Guard Extension (SGX)

Von Intel wurden kürzlich die Software Guard Extensions (SGX) [1] eingeführt und spezifiziert. Diese erlauben es, geschützte Softwarecontainer oder sogenannte *Enclaves* im laufenden System zu erstellen. Diese Container werden durch hardwareunterstützte Zugriffskontrollen geschützt. Dadurch wird es

ermöglicht, Enclaves von entfernten Gegenstellen bereitzustellen und trotzdem sicherzustellen, dass sensible Daten dem ungeschützten Bereich des Systems nicht preisgegeben werden. Die Details der Attestation-Verfahren bzw. Details über die Bereitstellung von Softwarecontainern sind in diesem Kontext unerheblich. Der Fokus wird im Rahmen dieses Projekts auf die Anwendbarkeit des Verfahrens zur vertrauenswürdigen Ausführung von Operationen auf möglicherweise sensiblen Daten gelegt. Enclaves können während des Betriebes eines Systems erstellt werden. Dabei ist es unerheblich, welches Betriebssystem ausgeführt wird. Applikationen, die in Enclaves ausgeführt werden, arbeiten ohne dabei ein eventuell unsicheres Betriebssystem einzubinden. Alle Schichten und Komponenten, die zur Abarbeitung des Programms benötigt werden, müssen in der Enclave ausgeführt werden. Auf den ersten Blick scheint Intel SGX also nur für kleine, in sich abgeschlossene Operationen auf kritischen Daten



geeignet zu sein. Bauman et al. [3] haben sich dieses Problems mit der Entwicklung von *Haven* angenommen. Dieses System startet in der Enclave eine eigene Windows-Betriebssystemschicht, die relevante Teile des Betriebssystems simuliert oder an das Host-Betriebssystem weiterleitet. Damit wird eine Ausführung von unmodifizierten Windows-Applikationen ermöglicht. Als Beispielapplikationen wurden ein Microsoft SQL Server und ein Apache Webserver erfolgreich in der abgeschirmten Umgebung ausgeführt. Die erzielten Ergebnisse können nicht ohne weiteres reproduziert werden, da einerseits Änderungen am Windows-Kernel (der nicht zur Verfügung steht) durchgeführt wurden, und andererseits an die Grenzen von Intel SGX gestoßen wurde. Es wurden Änderungen für zukünftige SGX-Iterationen vorgeschlagen, um diese Technologie auch für bestehende Applikationen nutzbar zu machen. Das Potential dieser Technologie ist groß, und kann prinzipiell dafür verwendet werden um in der Cloud, auf Servern, und Desktop-Computern, deren Vertrauenszustand unbekannt ist, sensible Daten zu verarbeiten.

3. Authentifizierung mit Identifikator

Weitaus häufiger wird Authentifizierung benötigt, die an einen Identifikator gebunden ist. Generell ist eine Identifikation von Entitäten ohne Hilfsmittel wohl nicht möglich. Ein Beispiel ist die Identifizierung einer Person in einem Gespräch. Ohne Hilfsmittel wie Register oder Dokumente gibt es keine Möglichkeit zu überprüfen, ob eine Person die Person ist, die sie angibt zu sein. Und selbst mit Dokumenten und Registern kann die Genauigkeit der Identifikation nur angenähert werden. Je mehr Informationen vorhanden sind, beispielsweise Unterschriftenbeispiel, Foto und Fingerabdruck, desto besser kann die Identität eingegrenzt werden. Eine theoretische 100%ige Sicherheit wird wohl nie erreicht werden. In der IT haben sich kryptografische Schlüssel als Äquivalent zu Dokumenten und Informationen bewährt und eingebürgert. Es ist damit möglich, einen kryptografischen Beweis für die Identität des Schlüssels zu führen. Offen bleibt aber, welcher Identität ein kryptografischer Schlüssel zugeordnet ist. In der IT hat sich das Konzept der Zertifizierungsdienste (englisch: Certification Authorities – CAs) als Bestätigung der Verbindung von Identität und Schlüssel etabliert. Das klassische Beispiel für eine Identifizierungsnotwendigkeit in der IT ist ein Web-Portal, das einem Benutzer bzw. einer Benutzerin personalisierte Services anbietet. Der Benutzer stellt über kryptografische Methoden – typischerweise ein TLS-Zertifikat – sicher, dass das aktuell verwendete Portal auch wirklich das gewünschte Portal ist. Zertifikate werden von Zertifizierungsdiensten ausgegeben und sind meist über die Domain an das Portal gebunden. Folglich besteht eine Abhängigkeit zum Domain Name System. Für neue Technologien wie WebRTC, Websockets, udgl. existiert aber kein Äquivalent zu DNS-Infrastrukturen. Prinzipiell gibt es zwei Möglichkeiten, Alternativen zu CA-basierten Authentifizierungsverfahren umzusetzen. Zum einen können unter bestimmten Voraussetzungen Authentifizierungsverfahren ohne eine zentrale vertrauenswürdige Instanz ausgeführt werden, z. B. indem ein Portal alleine durch einen korrekten Service seine Authentizität beweisen kann. Andererseits können CA-ähnliche Strukturen nachempfunden werden und so die Erfahrungen aus Public-Key-Infrastrukturen und Zertifizierungsdiensten genutzt werden. Im Folgenden werden einige alternative Konzepte zu klassischen CA-basierten Verfahren vorgestellt.

3.1. Zertifizierungsdienste

Ein klassischer Zertifizierungsdienst in einer Public-Key-Infrastruktur hat die Aufgabe, (auf Anfrage) Entitäten auf ihre Authentizität und Identität zu überprüfen. Stimmt die Identität, zertifiziert der Zertifizierungsdienst diese, indem es ein Zertifikat ausstellt das besagt, dass der angegebene Zertifizierungsdienst die Identität der Entität überprüft und für korrekt befunden hat und ein angegebener kryptografische Schlüssel zu dieser Identität gehört. Der Benutzer bzw. die Benutzerin muss sich somit auf den Zertifizierungsdienst verlassen. Im klassischen Anwendungsfall zur Authentifizierung eines Web-Portals reicht dies meist aus. Bedeutende Nachteile sind, dass Nutzer und Nutzerinnen meist keinen Einfluss auf die Qualität der Zertifizierung nehmen können und dass eine Zertifizierung je nach Qualität eine gewisse Zeit in Anspruch nimmt und somit für kurzlebige Dienste möglicherweise nicht rentabel ist. Zusammenfassend ist das Konzept von Zertifizierungsdiensten eine hinreichende Lösung für Authentifizierung von Webseiten und Portalen im Internet, solange klassische Kommunikationsprotokolle in meist statischen Umgebungen verwendet werden.

3.2. Web of Trust

Die wohl bekannteste Antithese zu traditionellen CA-Hierarchien sind so genannte *Webs of Trust*, wie sie beispielsweise bei *Pretty Good Privacy* (PGP) [5] zum Einsatz kommen. Hierbei wird asymmetrische Kryptografie eingesetzt und den Beteiligten sind einander ihre öffentlichen Schlüssel bekannt. Die Zuordnung eines Schlüssels zu einer Partei wird dabei von all jenen, welche diese überprüft haben bestätigt und diese Bestätigung formalisiert und digital signiert. Zusätzlich kann jeder Teilnehmer und jede Teilnehmerin festlegen, wem er oder sie in welchem Maß vertraut. Daraus lassen sich Vertrauensketten bilden, wodurch es beispielsweise möglich ist, dass zwei Parteien, welche einander nicht kennen, die Identität der jeweils anderen anhand deren öffentlichen Schlüssel verifizieren können. Voraussetzung hierfür ist, dass die Identität des Gegenübers indirekt, über „gemeinsame Bekannte", bestätigt werden kann. PGP definiert jedoch abgesehen von einem abstrakten Protokoll für die Zuordnung eines Schlüssels und die Bestätigung dieser Assoziation durch eine digitale Signatur keine Vorgehensweise, um diese Identität zu überprüfen. Tatsächlich ist es gängige Praxis, Schlüssel persönlich auszutauschen, abzugleichen und manuell zu bestätigen. Daher ist dieser Prozess auch nicht automatisierbar, ohne die Sicherheitseigenschaften von PGP abzuschwächen. Es existieren auch Protokolle, welche ohne (potentiell komplexe) soziale Netze und unterschiedliche Vertrauensstufen auskommen, um andere Teilnehmer, bzw. den Kommunikationskanal zu diesen Teilnehmern zweifelsfrei zu authentifizieren. Diese werden im Folgenden beschrieben.

3.3. Off-the-Record Messaging und SRTP/ZRTP

Eine Weiterentwicklung einiger Prinzipien von PGP findet sich in Protokollen wie *Off-the- Record Messaging* (OTR) [6] für Intant-Messaging und *Secure Real-time Transport Protocol* (SRTP) [2] bzw. ZRTP [7] für Voice– und Videochat. In beiden Fällen kommen Protokolle zum Einsatz, welche über die geschickte Kombination von asymmetrischer Kryptografie, symmetrischen Verschlüsselungsverfahren und *Key Agreement* sicherstellen können, dass ein direkter, abgesicherter Kommunikationskanal zwischen zwei Parteien hergestellt werden kann. Sobald ein derartiger Kanal aufgebaut wurde, müssen jedoch beide Kommunikationsteilnehmer ihr Gegenüber authentifizieren. Hierfür gibt es mehrere Möglichkeiten. Dies kann beispielsweise über einen Frage-Antwort-Prozess geschehen, im Zuge dessen dem Gegenüber eine Frage gestellt wird, welche nur von der Person beantwortet werden kann, mit der man glaubt zu kommunizieren. Bei korrekter Antwort kann man sich somit sicher sein, dass man tatsächlich mit der Person kommuniziert, mit der man glaubt in Kontakt zu sein. Da auf Protokollebene sichergestellt wird, dass tatsächlich ein direkter Kommunikationskanal aufgebaut wird, stellt sich sozusagen nur mehr die Frage mit wem – *Man-in-the-Middle*- Angriffe können protokollbedingt ausgeschlossen werden. Abgesehen von einer Frage-Antwort-Authentifizierung, welche voraussetzt, dass beide Kommunikationspartner über ein gemeinsames geheimes Wissen verfügen, können auch analog zu PGP die öffentlichen Schlüssel manuell einer Identität zugeordnet werden. Weiters kann auch ein im Zuge des Verbindungsaufbaus generierter Zufallswert über andere Kommunikationskanäle, wie z.B. SMS abgeglichen werden. Kommen SRTP/ZRTP für Voice-Chat zum Einsatz, wird dieser Prozess stark vereinfacht, da der Zufallswert dem Gegenüber direkt angesagt werden kann. Solange keine gezielte Manipulation natürlicher Sprache in so hoher Qualität zu Einsatz kommt, dass die Manipulation für das menschliche Gehör nicht mehr nachvollziehbar ist, ist dieser mündliche Abgleich ausreichend, um die Identität des Gegenübers, bzw. einen gesicherten, direkten Kommunikationskanal zu verifizieren.

3.4. Shared Secret

Das bekannte Passwort-Konzept, bzw. Abwandlungen davon, können auch für Authentifizierung von Entitäten verwendet werden. Ein Dienst kann beispielsweise so umgesetzt werden, dass dieser nur korrekte Ergebnisse liefert, wenn auch wirklich der korrekte Dienst angesprochen wird. Ein Web-Portal, das für den Benutzer kryptografische Schlüssel verwaltet, kann eine Entschlüsselungsaufgabe nur dann erfolgreich durchführen, wenn auch wirklich der korrekte Schlüssel zur Verfügung steht. Geht man davon aus, dass solches Schlüsselmaterial nicht gestohlen werden kann, ergibt sich so eine eindeutige, starke Authentifizierung des Portals gegenüber seinen Nutzern und Nutzerinnen. Die Schwachstelle in diesem Konzept liegt in der initialen Kontaktaufnahme zum Dienst: Wie kann beim erstmaligen Aufruf festgestellt werden, dass das korrekte Portal das Schlüsselmaterial generiert und mit dem Benutzer bzw. der Benutzerin austauscht. Um diese *Take-Ownership-Prozedur* erfolgreich durchzuführen, wird meist mit unabhängigen sicheren Kanälen ähnlich wie bei OTR gearbeitet. Diese und ähnliche Arten der Shared-Secret-Methodik können somit durchaus für Authentifizierung von Portalen oder Services eingesetzt werden. Konkrete Anwendungsfälle finden sich dabei vermehrt in modernen Infrastrukturen und weniger in klassischen Client-Server-Anwendungen.

3.5. Oauth

Eine Möglichkeit die bestehenden (digitalen) sozialen Netze und die von Benutzern und Benutzerinnen genutzten Online-Dienste und deren Profile für Identitätsnachweise heranzuziehen, wurde über *OAuth* [9] standardisiert. Sofern dies von den betreffenden Diensten unterstützt wird, ist es möglich, ein Kundenkonto eines Dienstes heranzuziehen, um sich einem anderen Dienst gegenüber zu identifizieren. Im Gegensatz zu den zuvor genannten Konzepten, kommen hier wieder zentrale, vertrauenswürdige Instanzen zum Einsatz. Diese sogenannten *Identity Provider* (IdP) erlauben es anderen Diensten, jene Informationen über einen Benutzer bzw. eine Benutzerin bekannt zu geben, welche zuvor explizit vom Benutzer oder der Benutzerin freigegeben wurden. Prinzipiell kann jeder Online-Dienst als IdP fungieren und es so seinen Mitgliedern erlauben, sich anderen Diensten gegenüber zu identifizieren. Durch diese Identifizierung über Dritte, müssen keine sensiblen Informationen wie z.B. Passwörter zum Serviceanbieter übermittelt werden, da der Identifikationsprozess ausgelagert wird. Obwohl in diesem Fall wieder zentrale Instanzen zum Einsatz kommen, ergeben sich entscheidende Vorteile gegenüber klassischen CA-Szenarien. Insbesondere ist kein aufwändiger Zertifizierungsprozess notwendig, um die Identität einer Partei festzustellen, sondern lediglich ein Austausch von Eckdaten über den IdP. Im Gegensatz zu PGP teilt sich dieser Ansatz einen Vorteil klassischer CA-Szenarien: Kommen Zertifizierungsstellen zum Einsatz, kann das Ausstellen von Zertifikaten auch mit bestimmten rechtlichen Garantien einhergehen und die Frage der Haftung im Falle eines Missbrauchs ist vergleichsweise einfach zu klären [4]. Ähnliche Vorgehensweisen sind auch für Identity Provider denkbar, da im Gegensatz zu PGP klar ist, wer für die Identifikation von Parteien verantwortlich ist, bzw. für Nachlässigkeit verantwortlich gemacht werden kann. Abgesehen davon gibt es noch einige stark an den Anwendungsfall gebundene Authentifizierungsmöglichkeiten, welche in den folgenden Abschnitten neben konkreter Anwendungsmöglichkeiten für die bisher beschriebenen Verfahren beschrieben werden.

4. Prototypen

4.1. Bewerten der Vertrauenswürdigkeit von Geräten

Es wurden in diesem Dokument drei Verfahren zur Bewertung der Vertrauenswürdigkeit beschrieben. Das Intel SGX-Verfahren kann hierfür die höchsten Sicherheitsgarantien anbieten, ist derzeit allerdings hinsichtlich der technischen Realisierbarkeit limitiert:

```
   - SGX-fähige Hardware ist derzeit auf Server-Systemen noch nicht verfügbar und
wird erst mit der nächsten Generation zur Verfügung stehen.

   - Desktopcomputer, die mit Intel SGX ausgestattet sind, bieten noch nicht die
notwendigen Voraussetzungen. Das von Bauman et al. [3] entwickelte System, um
unveränderte Applikationen auszuführen, benötigt außerdem Änderungen am SGX-
Befehlssatz. Des Weiteren ist diese Implementierung nicht frei verfügbar, sondern
benötigt auch Änderungen an Closed-Source Modulen wie dem Windows-Kernel. Dieses
Verfahren bleibt also vorerst ein theoretisches Konzept, kann aber in naher Zukunft
Anwendung bei sicheren Cloudanwendungen finden. Das SafetyNet-Verfahren von Google
wurde, wie bereits in Abbildung 1 dargestellt, ohne die Verwendung von mehreren
Geräten umgesetzt. Dabei werden Geräte mit unmodifizierten Android-Installationen
erfolgreich als vertrauenswürdig erkannt. Prinzipiell werden gerootete Geräte auch
als solche erkannt, allerdings kann diese Root-Erkennung, wie bereits angedeutet,
umgangen werden, etwa über _suhide_ [6] . Erst kürzlich wurde von Google eine
Aktualisierung des Safetynet Dienstes durchgeführt, um eine bessere Erkennungsrate
zu erhalten. Mittlerweile wurden aber suhide ebenfalls nachgebessert. Der MDM Fall
wurde nicht explizit nachgebildet, da dazu kommerzielle Software und ein
erheblicher Konfigurationsaufwand notwendig sind. Prinzipiell leiden aber alle rein
Software-basierten Ansätze unter denselben Schwächen wie der SafetyNet-Ansatz. Dem
kann nur im Falle von von Firmen kontrollierten Smartphones über entsprechende
organisatorische Policies entgegengewirkt werden.
```

4.2. Sicherer Informationsaustausch auf Basis des Inverted Trust Model

Systeme wie PGP ermöglichen nur dann eine gesicherte Informationsübertragung, wenn der öffentliche Schlüssel des Gegenübers bekannt ist und diesem zweifelsfrei zugeordnet werden kann. Eine konzeptionelle Unzulänglichkeit dieses Ansatzes liegt darin begründet, dass alle Beteiligten ihr eigenes Schlüsselmaterial generieren und verwalten. Selbst wenn nur der Empfänger bzw. die Empfängerin problematische kryptografische Verfahren einsetzt, welche in einem Schlüsselpaar resultieren, welches keinen ausreichenden Schutz bietet, kann für die Sicherheit der übertragenen Daten nicht garantiert werden. Abhilfe schafft ein *Inverted Trust Model* [8] bei dem sich die Partei, welche die Daten empfängt, gegenüber dem Sender authentifizieren muss, um Zugriff auf die zuvor empfangenen Daten zu erhalten. Aus diesem Umstand ergeben sich einige Vorteile: Zum einen können die Daten selbst an beliebiger Stelle ausgelagert werden, da diese ohnehin vor unautorisierten Zugriffen geschützt sind – erst bei erfolgreicher Authentifizierung wird Zugriff erteilt. Ein weitaus wichtigerer Aspekt dieses Ansatzes ist jedoch, dass die Instanz, welche die Daten zur Verfügung stellt, alles nötige Schlüsselmaterial selbst verwaltet und daher durch den Einsatz entsprechender kryptografischer Methoden direkt festlegen kann „wie sicher“ die zu

übertragenden Daten abgespeichert werden sollen. Somit liegt die volle Kontrolle auch am Ursprungsort der zu schützenden Daten. Eine weitere Konsequenz aus einem System, welches Empfänger und Empfängerinnen dazu verpflichtet, sich zu authentifizieren, ist, dass Sender anonym bleiben können und trotzdem selbst über die eingesetzten kryptografischen Verfahren und das gesamte Schlüsselmaterial entscheiden. Im Speziellen können auch je nach Anwendungsfall und Anforderungen an Datensicherheit, Sender-Anonymität und Identitätsnachweise, unterschiedlich starke oder schwache Authentifizierungsmaßnahmen zum Einsatz kommen. Der Datenaustausch läuft, unter Zuhilfenahme des Inverted Trust Model, wie folgt ab:

```
   - Der Sender generiert Einweg-Schlüsselmaterial und verschlüsselt Daten für den
Empfänger.

   - Der Empfänger authentifiziert sich gegenüber dem Sender, um Zugriff auf den
Einwegschlüssel zu erhalten.

   - Nach erfolgter Authentifizierung wird das Schlüsselmaterial übertragen.
```

6 http://forum.xda-developers.com/apps/supersu/suhide-t3450396

```
   - Der Empfänger entschlüsselt die Daten mit dem eben empfangenen Schlüssel.
Abbildung 2 veranschaulicht diesen Ablauf. Auf die Details zum
Authentifizierungsvorgang wird in dieser Abbildung bewusst verzichtet, da beliebige
Verfahren zum Einsatz kommen können. Wichtig ist in diesem Zusammenhang jedoch,
dass die Kanäle, welche zum Austausch des Schlüsselmaterials und zur
Authentifizierung verwendet werden, abgesichert sein müssen. Eine Möglichkeit, dies
umzusetzen, besteht darin, Mehrfaktor-Authentifizierung in Kombination mit OTR zu
verwenden, um Man-In-the-Middle-Angriffe auszuschließen und gleichzeitig den
Empfänger zu authentifizieren. Eine andere Variante dieses Konzept praktisch
umzusetzen ist, einen Identity Provider für
```

Abbildung 2: Authentifizierungsablauf im Rahmen des Inverted Trust Model die Authentifizierung heranzuziehen. Damit ergeben sich auch keine Probleme, gesicherte, authentifizierte Kanäle zwischen den Kommunikationspartnern aufzubauen, wenn beide Parteien dem IdP vertrauen:

- Der Sender registriert sich bei einem IdP und bindet die Information über den IdP an die zu übertragenen Daten, sodass diese Information vom Empfänger ausgelesen werden kann.

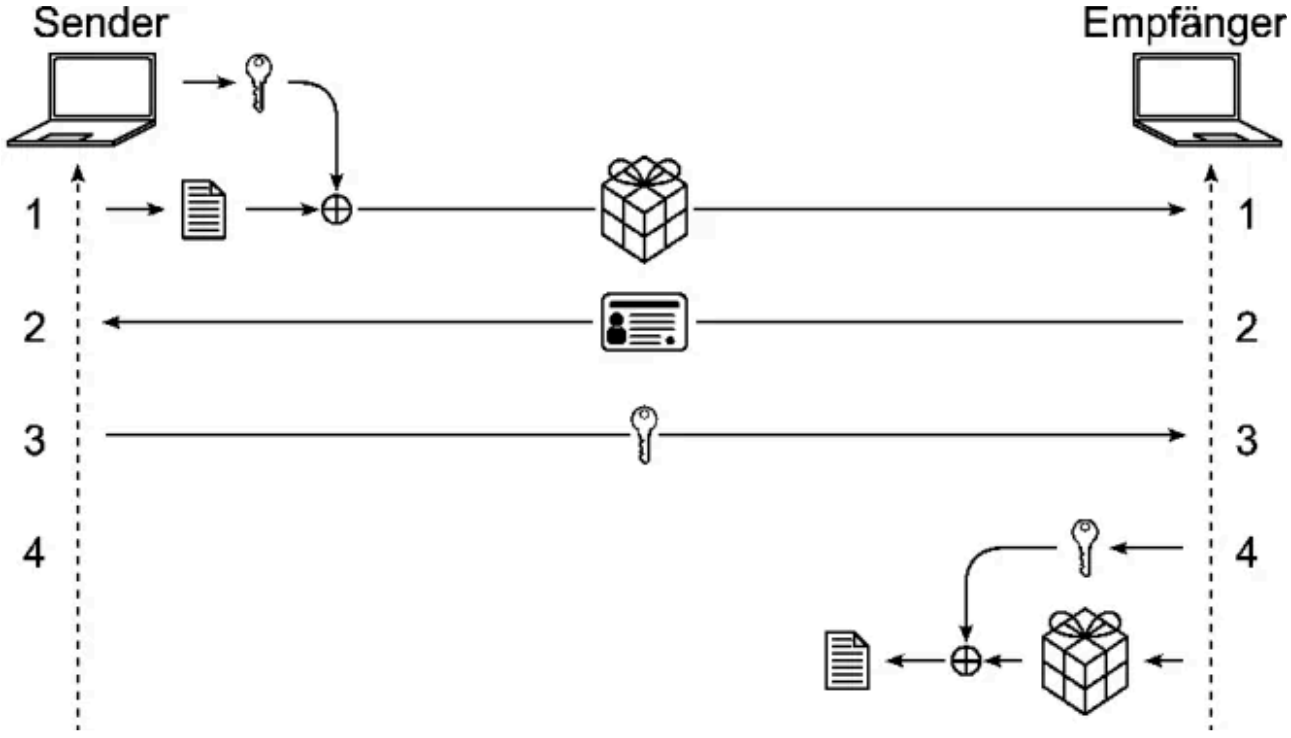
- Sobald der Empfänger sich zu authentifizieren versucht, wird dieser an den IdP weitergeleitet.

- Der Empfänger identifiziert sich gegenüber dem IdP – für diesen Schritt können wieder CA-basierte Ansätze herangezogen werden, da ein IdP typischerweise langlebig ist.

- Der IdP informiert den Empfänger nach erfolgter Authentifizierung, wie dieser den Sender erreichen kann, um Zugriff auf das notwendige Schlüsselmaterial zu erhalten. Diese Kontaktinformation kann vom Sender so gewählt werden, dass diese nur dem Empfänger und dem IdP bekannt ist – dem Empfänger also nur nach erfolgter Authentifizierung bekannt gemacht wird.

- Der Empfänger kann sich sicher sein, dass die Kontaktinformation korrekt ist, so lange sich der IdP korrekt verhält.

- Wird der Sender über die hinterlegte Kontaktinformation kontaktiert, kann dieser sicher sein, dass es sich um einen authentifizierten Benutzer handelt, und das Schlüsselmaterial kann übertragen werden. Zusätzlich kann über den IdP Information zur Verfügung gestellt werden, welche Aufschluss darüber gibt, welcher Benutzer zum Sender Kontakt aufnimmt, um eine Zugangskontrolle umzusetzen. Unabhängig von der verwendeten Authentifizierungsmethode muss sich der Sender gegenüber dem Empfänger nicht identifizieren bzw. authentifizieren. Wendet man die



eben beschriebenen Konzepte auf Szenarien in Zusammenhang mit Cloud-Diensten an, stellt man fest, dass man ähnliche Eigenschaften erhält, wie im Fall klassischer CAbasierter Authentifizierung: Auch hier wird die Authentizität des Benutzers (was dem Sender im obigen Beispiel entspricht) nicht auf Protokollebene überprüft. Je nach Art des genutzten Dienstes kann dies zu einem späteren Zeitpunkt auf Applikationsebene erfolgen. Umgelegt auf das Inverted Trust Model deckt sich dies mit der Eigenschaft, dass die übermittelten Daten für sich selbst sprechen (müssen) und die Authentizität der Daten aus diesen hervorgehen kann bzw. muss. Der Prototyp ist in das Open-Source Projekt CrySIL [7] integriert. CrySIL- der Cryptographic Service Interoperability Layer – konsolidiert und verbessert die Ergebnisse aus den Skytrust Projekten. An einer Demonstrationsplattform für die vielen Anwendungsfälle von CrySIL wird gerade gearbeitet.

5. Zusammenfassung

Im Zuge des Projekts wurden verschiedene existierende Technologien gegen die Anforderungen moderner Infrastrukturen und Anwendungsfälle evaluiert. Einerseits wurden Technologien diskutiert, die nur Authentifizierung ohne Identifizierung ermöglichen, andererseits wurden auch klassische Authentifizierungsansätze mit Identifizierung behandelt. Es stellte sich heraus, dass es in sehr eingeschränkten Anwendungsfällen mit speziellen Voraussetzungen sehr wohl möglich ist, ausreichend starke Authentifizierung zu erreichen. Allerdings stößt man an einem gewissen Punkt unweigerlich auf die Barriere der Identifikationsproblematik, die auch im nicht-technischen Bereich nicht vollständig lösbar scheint. Es bleibt somit großteils offen, Authentifizierungsanforderungen moderner Technologien zu befriedigen.

Literatur

[1] Ittai Anati u. a. “Innovative Technology for CPU Based Attestation and Sealing”. In: HASP Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy (2013), S. 1–7. DOI: 10.1.1.405.8266 .

[2] M. Baugher u. a. The Secure Real-time Transport Protocol (SRTP). RFC 3711. März 2004.

URL: https://tools.ietf.org/html/rfc3711 (besucht am 25. 11. 2016).

[3] Andrew Baumann, Marcus Peinado und Galen Hunt. “Shielding Applications from an Untrusted Cloud with Haven”. In: Proceedings of the 11th USENIX Conference on Operating Systems Design and Implementation (2014), S. 267–283. ISSN: 0734-2071. DOI: 10.1145/2799647 . URL: http://dl.acm.org/citation.cfm?id=2685048.2685070.

[4] Johannes A. Buchmann, Evangelos Karatsiolis und Alexander Wiesmaier. Introduction to Public Key Infrastructures. Berlin, Heidelberg, Germany: Springer-Verlag, 2013. ISBN: 978-3-642-40656-0.

[5] Simson Garfinkel. PGP: Pretty Good Privacy. Sebastopol, CA, USA: O’Reilly & Associates, 1995. ISBN: 1-56592-098-8.

[6] Ian Goldberg und the OTR Development Team. Off-the-Record Messaging Protocol version 3.

URL: https://otr.cypherpunks.ca/Protocol-v3-4.0.0.html (besucht am 06. 01. 2016).

[7] A. Johnston, P. Zimmermann und J. Callas. ZRTP: Media Path Key Agreement for Unicast Secure RTP. RFC 6189. Apr. 2011. URL: https://tools.ietf.org/html/rfc6189 (besucht am 25. 11. 2016).

[8] Florian Reimair, Peter Teufl und Bernd Prünster. “In Certificates We Trust - Revisited”. In:

Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. 2015, S. 702–709.

[9] The OAuth Community. OAuth Community Site. 2015. URL: http://oauth.net (besucht am 18. 11. 2015).

7 https://github.com/IAIK/CrySIL