Jonas Lehner, Andreas Oberweis, Gunther Schiefer

Kontextabhängige 3-Faktor-Authentifizierung für den mobilen Zugriff auf Unternehmensanwendungen

Der Zugriff von mobilen Geräten auf Unternehmensanwendungen birgt Sicherheitsrisiken, die über die stationärer Hardware hinausgehen. Diese umfassen unter anderem den Verlust oder den Diebstahl ganzer Geräte, das Aushorchen der drahtlosen Internetverbindung oder das Ausspähen per Shoulder-Sniffing. Im Rahmen des Projektes „SumoDacs“ wurde eine 3-Faktor-Authentifizierung entwickelt, die neben einem konventionellen Zugangspasswort aus einem Hardware-Vertrauensanker und einer kontextsensitiven Zugriffskontrolle besteht. Die Authentifizierung wird clientseitig durch einen lokalen Proxy-Dienst realisiert.

Inhaltsübersicht

1 Risiken durch mobilen Zugriff auf Unternehmensanwendungen

Im Kontext aktueller Entwicklungen, wie Bring Your Own Device (BYOD) und der wachsenden Mobilität der Mitarbeiter, spielt der Zugriff auf Unternehmensdaten außerhalb des eigenen (kontrollierbaren) Firmengeländes eine zunehmend wichtigere Rolle. Neben den aus dem stationären Betrieb von IT-Systemen bekannten Sicherheitsrisiken ergeben sich dadurch zusätzliche Angriffsszenarien. Diese umfassen unter anderem den Verlust oder den Diebstahl ganzer Geräte (Smartphone, Laptop etc.), das Aushorchen der meist drahtlosen Internetverbindung oder das sog. Shoulder-Sniffing, also das optische Ausspähen von

Informationen während der Benutzung durch den Mitarbeiter. Da der Einsatz mobiler Geräte erst dann einen wirklichen Mehrwert bietet, wenn diese dazu genutzt werden, mit aktuellen Unternehmensdaten zu arbeiten und nicht etwa nur zur mobilen (Offline-)Datenerfassung, muss mit diesen Risiken angemessen umgegangen werden.

1.1 Mobiler Datenzugriff auf Unternehmensdaten

Da die mobile Konnektivität zunehmend besser wird, ist ein Ansatz, um die Risiken zu verringern und gleichzeitig mit immer aktuellen Daten zu arbeiten, der jederzeitige Zugriff über drahtlose Netze auf die Unternehmensdaten. Um dadurch keine neuen Sicherheitslücken zu schaffen, muss der Zugriff auf die Daten sehr gut kontrolliert werden. Dazu muss der Zugriff nicht nur beim ersten Mal, sondern permanent kontrolliert werden. Hierzu reicht es nicht, nur eine einmalige Autorisierung zu Beginn einer Sitzung durchzuführen. Möglich wäre eine regelmäßige erneute Überprüfung der Zugangsdaten. Um ein ausreichend gutes Sicherheitsniveau zu erreichen müsste die Frequenz der Wiederholung sehr hoch gewählt werden. Dies würde vermutlich in vielen Fällen dazu führen, dass die Nutzer sich alternative Zugangsmöglichkeiten suchen, welche dann in der Regel weniger gut abgesichert sind. Deshalb wird hier ein Verfahren beschrieben, welches zusätzlich zu den Zugangsdaten (diese muss der Nutzer *wissen*) ein mit dem Gerät verbundenes Sicherheits-Token voraussetzt (dieses muss der Nutzer *haben*). Damit kann jede einzelne Datenanfrage ohne Nutzerinteraktion autorisiert werden. Weiterhin können jeder Anfrage definierte Kontextparameter mitgegeben werden, welche vor Verarbeitung der Anfrage auf Gültigkeit - entsprechend vorgegebener Regeln - geprüft werden. Der Nutzer muss somit z.B. an einem definierten Ort *sein* bzw. nicht sein.
Dadurch dass die Daten immer vom Unternehmensserver abgerufen werden, sind sie bei mobiler Nutzung genauso aktuell wie am Arbeitsplatz im Unternehmen. Sie werden nicht längerfristig auf dem Mobilgerät gespeichert und können damit nicht in falsche Hände geraten, falls das Mobilgerät gestohlen wird oder verloren geht. Die Notwendigkeit zum Schutz der Daten (bei der mobilen Nutzung) reduziert sich auf die (wenigen) aktuell im Speicher befindlichen Daten, die Absicherung des Datentransports und eine zuverlässige Kontrolle über den berechtigten Datenzugriff.

1.2 Aktuelle Lösungen für den mobilen Datenzugriff

Veröffentlichungen über die Authentifizierung von Nutzern mittels Smartcards gibt es viele (z.B. [Yang et al. 1999]). Die meisten Verfahren dienen nur der ersten Authentifizierung bei einer Sitzung. Hier können auch nur kurzzeitig verbundene Smartcards wie z.B. der neue Personalausweis eingesetzt werden. Ein Beispiel für eine permanent aktive Smartcard ist das *Subscriber Identity Module (SIM)*. Die SIM-Karte dient (u.a.) der permanenten Authentifizierung in Mobilfunknetzen. Da sie die jedoch immer dem Mobilfunkanbieter gehört und nicht alle mobilen Geräte über Mobilfunknetze kommunizieren, ist es nicht möglich diese universell einzusetzen.
In vielen Fällen werden *Virtuelle Private Netzwerke (VPN)* eingesetzt, um mobile Geräte in das eigene private Unternehmensnetzwerk einzubinden. Diese Lösung dient jedoch nur der Transportverschlüsselung durch eine einmalige Authentifizierung (normalerweise nur Nutzer/Passwort) zu Sitzungsbeginn, eine permanente Authentifizierung findet nicht statt.
Eine andere Möglichkeit ist der Einsatz von biometrischen Informationen. Allerdings gestaltet es sich schwierig, hier eine permanente Überprüfung umzusetzen. Darüber hinaus haben biometrische Informationen das Problem, dass diese nicht gegen neue ausgetauscht werden können, sollten sie kompromittiert worden sein.
Mobile Device Management (MDM) ist ein verbreiteter Ansatz um mehr Kontrolle über die mobilen Geräte der Nutzer von Unternehmensdaten zu haben. Diese Softwarelösungen dienen im Allgemeinen zur Absicherung der Geräte gegen unerwünschte Nutzungen (auch durch den

Nutzungsberechtigten) oder zur entfernten Datenlöschung. Eine Absicherung des Datenzugriffs beinhalten MDM-Lösungen üblicherweise nicht.
Verschlüsselte Datenspeicherung auf den Geräten liefert auch nicht die gewünschte Sicherheit, da diese Daten entweder nicht aktuell sind oder beim Aktualisieren wieder die bekannten Sicherheitsprobleme auftreten.
Keiner dieser Lösungsansätze unterstützt eine kontextsensitive Zugriffskontrolle, wie sie im folgenden Verfahren eingesetzt wird.

2 Gesamtarchitektur

Im Rahmen des vom Bundesministerium für Bildung und Forschung geförderten Projektes „SumoDacs“ [WIBU 2011] wurde für den mobilen Zugriff auf Unternehmensdaten eine 3-Faktor-Authentifizierung entwickelt, die neben einem konventionellen Zugangspasswort aus einem sog. Hardware-Vertrauensanker (Smartcard) und einer kontextabhängigen Zugriffskontrolle besteht. Zunächst findet eine Überprüfung der Identität der Unternehmensanwendung statt bevor eine verschlüsselte Verbindung aufgebaut wird. Zur Authentifizierung des Nutzers und zur Gewährung des Zugriffs muss neben der richtigen Kombination von Benutzername und Passwort auch die korrekte Signatur des Hardware-Tokens im Challenge-Response-Verfahren vorliegen. Jede Anfrage, die anschließend über die verschlüsselte Ende-zu-Ende-Verbindung übertragen wird, muss wiederum von dem im Hardware-Token sicher abgelegten privaten Schlüssel signiert sein. Darüber hinaus prüft die kontextabhängige Zugriffskontrolle, ob Kontextinformationen vorliegen, die der Anfrage widersprechen bzw. diese erst zulässig machen. Abb. 1 gibt einen Überblick über die Architektur.

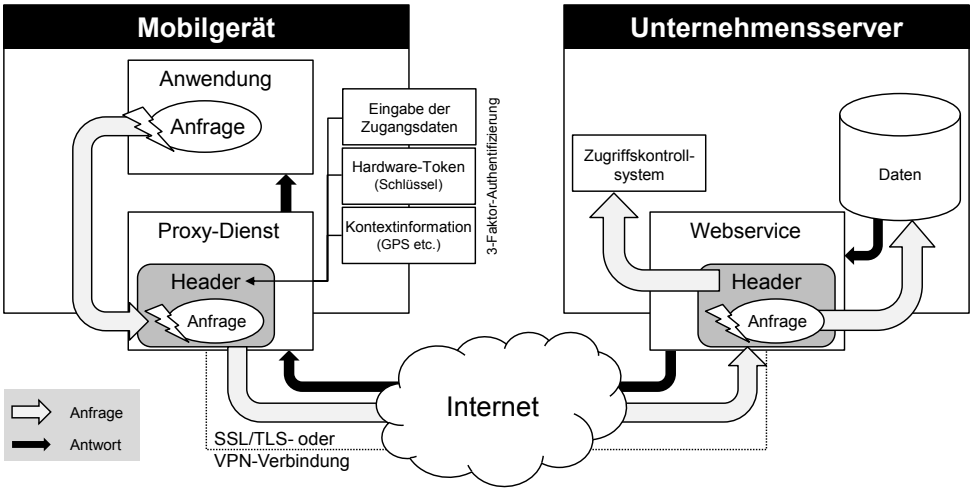

Abb. 1: Gesamtarchitektur

Die Authentifizierung wird technisch durch einen lokalen Proxy-Dienst realisiert, über den alle Anfragen an die Unternehmensanwendung umgeleitet werden. Die notwendigen Zugriffsinformationen werden dem Header der Anfrage hinzugefügt, sodass der Unternehmensserver diese prüfen kann.
Das Konzept beruht darauf, dass die Authentifizierungsinformationen an jede Anfrage der Anwendung angehängt werden. Diese sind bewusst nicht in die Anfrage integriert, da dies die Umsetzungsmöglichkeiten einschränken würde. Es ist somit möglich, das entwickelte Konzept

zwischen eine bestehende mobile Anwendung und eine bestehende Unternehmensanwendung zu setzen. Es kann jedoch auch auf einer oder beiden Seiten in die jeweilige Anwendung integriert werden. Im Projekt SumoDacs wurde auf Seiten der mobilen Anwendung ein Standardbrowser mit einer separaten Proxy-Anwendung verwendet, auf Seiten der Unternehmensanwendung wurden die Sicherheitskomponenten in die Anwendungen CAS-PIA der CAS Software AG integriert [CAS 2013].

3 Hardware-Sicherheits-Token

Im Projekt SumoDacs wurde für die Sicherheitstoken die CodeMeter®-Technologie der WIBU-SYSTEMS AG verwendet. Die Entwicklung stammt aus dem Software- und Lizenzschutz und ist in verschiedenen Bauformen verfügbar (z.B. USB, SD, µSD).
Die Technologie bietet alle nötigen Funktionen, um den Zugang zu Web-Anwendungen und zu Software-as-a-Service-Lösungen abzusichern. Der private Schlüssel, als eindeutiges Merkmal für den Benutzer, wird sicher im Hardware-Token gespeichert. Durch den Einsatz von asymmetrischer Kryptographie (Public / Private Keys) wird für die nötige Sicherheit gesorgt.

3.1 Grundsätzliche Funktion

Im Sicherheits-Token wird ein privater *Elliptic Curve Cryptography (ECC)* Schlüssel [Miller 1985] mit 224 Bit Schlüssellänge gespeichert. Ein 224 Bit ECC Schlüssel entspricht sicherheitstechnisch einem 2048 Bit RSA Schlüssel [NSA 2009]. Der zugehörige öffentliche Schlüssel ist beim Unternehmensserver hinterlegt. Beim Anmelden erzeugt der Unternehmensserver eine zufällige Zeichenkette (Challenge) und sendet diese an den Client. Dieser signiert die Challenge mit dem privaten Schlüssel und schickt sie mit Signatur als Antwort (Response) an den Unternehmensserver zurück. Mittels des öffentlichen Schlüssels überprüft der Unternehmensserver die Identität des Anwenders. Da der Unternehmensserver nur den öffentlichen Schlüssel kennt, kann ein Angreifer - selbst wenn er diesen Schlüssel entwendet - sich nicht als Anwender ausgeben, da der private Schlüssel nur im Sicherheits-Token gespeichert ist, welcher in den Händen des Mobilgerätenutzers ist.

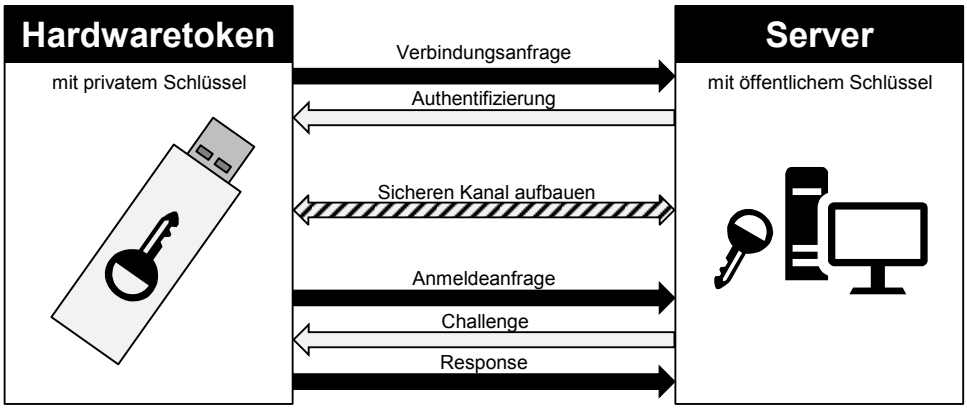

Abb. 2: Zugangsschutz mit privatem und öffentlichem Schlüssel

3.2 Authentifizierung

Bei der Verbindungsaufnahme des Mobilgerätes mit dem Unternehmensserver wird im ersten Schritt die Authentizität des Unternehmensservers überprüft. Der zweite Schritt ist die Etablierung

einer sicheren Kommunikation mit den üblichen Mitteln wie z.B. VPN-Verbindung oder SSL/TLS. Im dritten Schritt meldet sich das Mobilgerät beim Unternehmensserver an. Dazu wird das oben beschriebene Challenge-Response-Verfahren verwendet (vgl. Abb. 2). Die Response wird im Sicherheits-Token zwischengespeichert und kann so an jede folgende Anfrage angefügt werden. Zur Erhöhung der Sicherheit kann das Authentifizierungsverfahren in regelmäßigen Zeitabständen wiederholt werden, damit nicht nur für jede neue Sitzung geänderte Authentifizierungsinformationen vorliegen. Dies geschieht unbemerkt für den Nutzer, da keine Nutzerinteraktion nötig ist.

3.3 Vorteile des Sicherheits-Tokens

Durch den Einsatz des Sicherheits-Tokens kann allein durch Kenntnis von Benutzername und Passwort kein Zugriff auf die Unternehmensdaten erfolgen. Passwort-Sniffing oder Phishing führen für einen Angreifer nicht zum Erfolg. Ebenso können die privaten Authentifizierungsschlüssel nicht vom Unternehmensserver gestohlen werden, da dort nur die öffentlichen Schlüssel vorliegen. Durch den Einsatz der separaten Sicherheits-Token wird der Zugriff nicht auf dedizierte Geräte beschränkt. Eine vorhandene Client-Proxy-Komponente zusammen mit einem Sicherheits-Token ermöglichen den Zugriff. Da der private Schlüssel ggfs. auch auf mehrere Sicherheits-Token in unterschiedlichen Bauformen übertragen werden kann, stehen der flexiblen Nutzung der Unternehmensdaten durch den Nutzer ggfs. nur die Hard- und Softwarebeschränkungen der Gerätehersteller im Weg.
Die Nutzung der Sicherheits-Token ist für den Anwender sehr einfach. Diese müssen nur mit dem Mobilgerät verbunden (eingesteckt) werden und es muss (wenn gewünscht) ein Zugriffscode (PIN) eingegeben werden. Alles Weitere läuft für den Anwender unbemerkt ab.

3.4 Schlüsselgenerierung und -verteilung

Die Schlüsselgenerierung und -verteilung gestaltet sich sehr einfach. Durch die Verwendung der etablierten CodeMeter®-Technologie stehen die Werkzeuge dazu schon fertig zur Verfügung. Jedes Unternehmen bekommt ein spezielles Sicherheits-Token mit erweiterten Möglichkeiten, die sogenannte „Firm Security Box (FSB)“. Diese enthält einen eindeutigen Firmencode (Firm Code). Nur mit einer FSB können in den anderen Sicherheits-Token die privaten Schlüssel erzeugt werden. Auch das Übertragen eines privaten Anwenderschlüssels auf ein weiteres Sicherheits-Token ist nur in Verbindung mit einer FSB möglich. Die FSB verbleibt sicher verwahrt im Unternehmen. Mit Hilfe der Code Meter License Central [WIBU 2013] können private Schlüssel auch zuverlässig abgesichert online verteilt werden, eine direkte physische Zusammenführung von FSB und Sicherheits-Token ist damit nicht mehr nötig.

4 Kontextabhängige Zugriffskontrolle

Als weiterer Faktor zur Absicherung des Datenzugriffs kommen die kontextabhängigen Zugriffskontrollen hinzu. Diese dienen nicht der Authentifizierung des Anwenders sondern beschränken bzw. erlauben den Zugriff auf die Unternehmensdaten als Ganzes und/oder den Zugriff auf Teile der Daten ggfs. bis hin zu einzelnen Datenfeldern. Die möglichen Kontextparameter sind nicht beschränkt. Prinzipiell kann jede Kontextinformation verwendet werden, welche zur Laufzeit in maschinenlesbarer Form zur Verfügung steht [Dey 2001]. Verwendete Kontextparameter können z.B. der Ort, die Zeit oder Kalendereinträge sein. So kann beispielsweise eine Mitarbeiterin des Pflegedienstes im Krankenhaus nur im Zimmer des Patienten oder im Stationszimmer auf die Daten eines Patienten zugreifen. Der Zugriff von außerhalb des Krankenhauses ist ihr nicht gestattet. Ebenso ist die Wahrscheinlichkeit,

unbeabsichtigt eine Patientenakte zu verwechseln, deutlich reduziert, da der Zugriff aus anderen Patientenzimmern heraus nicht möglich ist (ein entsprechend genaues Ortungsverfahren vorausgesetzt).

4.1 Zugriffskontrollmodelle

Aufgabe einer Zugriffskontrolle ist es zu entscheiden, ob die Anfrage eines Anwenders zur Ausführung einer bestimmten *Operation* mit einem bestimmten *Objekt* zulässig ist. Beispiele für Objekte sind Dateien, Datenbanken, Datenfelder und auch Ressourcen wie Drucker, Festplatten, Webservices und dergleichen. Übliche Operationen sind z.B. lesen, schreiben, löschen, ändern usw. Der Anwender ist dabei das aktive Element und wird *Subjekt* genannt. Eine *Berechtigung* verknüpft ein Objekt mit einer Operation. *Zugriffskontrollmodelle (ZKM)* legen für jedes Subjekt fest, welche Berechtigungen es hat.
Im Folgenden werden die drei wichtigsten generischen Zugriffskontrollmodelle kurz skizziert [Ferrari 2010]. Daneben gibt es noch viele weitere aufgabenspezifische Modelle.

Mandatory Access Control (MAC)
Bei MAC handelt es sich um eine Systemvariante, die vor allem von Hochsicherheitseinrichtungen, wie Militär oder Geheimdiensten, verwendet wird. Das Prinzip beruht auf verschiedenen Vertraulichkeitsstufen (bspw. *Öffentlich*, *Vertraulich*, *Geheim*, *Streng Geheim*), die sowohl für Ressourcen als auch für die Subjekte (Nutzer) vergeben werden. Vereinfacht ausgedrückt darf ein Nutzer, der sich auf der Stufe *Geheim* befindet, auf Objekte der Stufen *Öffentlich*, *Vertraulich* und *Geheim* zugreifen, während ihm der Zugriff auf Objekte der Sicherheitsklasse *Streng Geheim* verwehrt wird.

Discretionary Access Control (DAC)
Das Grundprinzip dieses Zugriffskontrollmodells besteht darin, dass ein Subjekt (bspw. Nutzer), welches ein Objekt (bspw. Daten) erzeugt hat, über dessen Rechte verfügt und diese nach eigenem Ermessen an andere Subjekte weitergeben kann. Die Nutzungsrechte werden dabei für verschiedene Operationen, wie lesen, schreiben oder ausführen, vergeben. Mehrere Subjekte können zu Gruppen zusammengefasst werden, um die Zuordnung von Rechten zu vereinfachen. Eine Gruppe tritt dabei im System ebenfalls als Subjekt auf.

Role-Based Access Control (RBAC)
Im Gegensatz zu Nutzern und Gruppen bei DAC werden Berechtigungen bei RBAC ausschließlich über Rollen vergeben. Rollen entsprechen dabei Aufgabenbeschreibungen für Stellen und Positionen in Organisationen (bspw. „Vertriebsleiter“, „Vorstandsmitglied“, „Praktikant“) und sind mit den für die Aufgabenerfüllung notwendigen Rechten ausgestattet. Berechtigungen können bei RBAC nur über Rollen und nicht an einzelne Nutzer vergeben werden.

Im Projekt SumoDacs wurde das (einfachere) DAC-Modell gewählt. DAC ist leichter zu implementieren als RBAC und das in der Praxis vorherrschende ZKM. Das DAC-Modell wurde dann um Kontextsensitivität erweitert.
Kontext ist immer individuell für einen spezifischen Nutzer bzw. ein spezifisches mobiles Gerät. Da im Projekt Berechtigungen kontextabhängig geändert werden müssen, ist es notwendig einzelne Nutzer zu betrachten und nicht nur Rollen. Beispielsweise soll für einen Entwickler im Urlaub der Zugriff auf geheime Projektdaten nicht möglich sein, jedoch muss der Zugriff für alle anderen Entwickler unabhängig davon weiterhin gewährleistet sein. Es kann somit nicht der Zugriff für die Rolle „Entwickler“ gesperrt werden.

4.2 Kontextschalter

Kontextsensitivität bedeutet, dass eine Berechtigung von einer oder mehreren Kontextinformationen abhängt. Unter Kontext wird dabei jede Form von zur Laufzeit in expliziter Form für das System verfügbarer Information verstanden, mit der eine dynamische Anpassung an die aktuelle Situation des jeweiligen Nutzers vorgenommen werden kann.
Unter einem *Kontextschalter* wird eine Komponente in einen ZKM verstanden, die in Abhängigkeit von bestimmten Kontextparametern nach vorgegebenen Regeln dynamisch zur Laufzeit des Systems ein- und ausgeschaltet werden kann. Es kann sich hierbei um Entitäten handeln, aber auch um Assoziationen, mit denen Entitäten einander zugeordnet werden.

Abb. 3 zeigt das allgemeine kontextabhängige DAC-Modell für den Datenzugriff in SumoDacs. Es wurden zwei Stellen im Model identifiziert, die besonders dafür geeignet sind, als Andockpunkte für Kontextschalter zu fungieren:

- Operationale Berechtigung
  Unter einem operationalen Recht wird das grundlegende Recht für die Ausführung einer bestimmten Operation verstanden. Hier wird festgelegt, dass ein Subjekt für alle Objekte eine Berechtigung hat (oder nicht hat). Diese kann abhängig von der aktuellen Kontextsituation ein- oder ausgeschaltet werden. Beispiel: Bei einem mobilen Zugriff außerhalb des unternehmenseigenen WLAN ist keine Leseberechtigung gegeben, es können nur Daten (z.B. abgelesene Werte) in eine Datenbank geschrieben werden.

- Objektberechtigung
  Im Allgemeinen sind bei einem Datenzugriff die Objekte Datensätze (oder ggfs. auch einzelne Datenfelder). Objekte können aber auch aus anderen Objekten zusammengesetzt sein, bzw. diese enthalten, z.B. eine digitale Kundenakte mit allen relevanten Daten zu einem Kunden oder einzelne Termineinträge in einem Kalender. Mit Kontextschaltern für bestimmte Objekte können Operationen mit besonders sensitiven Daten unter bestimmten Bedingungen untersagt werden, dies kann auch den lesenden Zugriff beinhalten. Durch das Modell kann nicht nur der Zugriff auf einzelne Datensätze, sondern auch auf ganze Typen von Datensätzen kontrolliert werden. Beispielsweise darf ein Mitarbeiter im Außendienst generell nicht auf Personalakten zugreifen.

Durch die Möglichkeit, an verschiedenen Komponenten des Berechtigungsmodells Kontextschalter anzubringen gewinnt das Modell an Flexibilität.

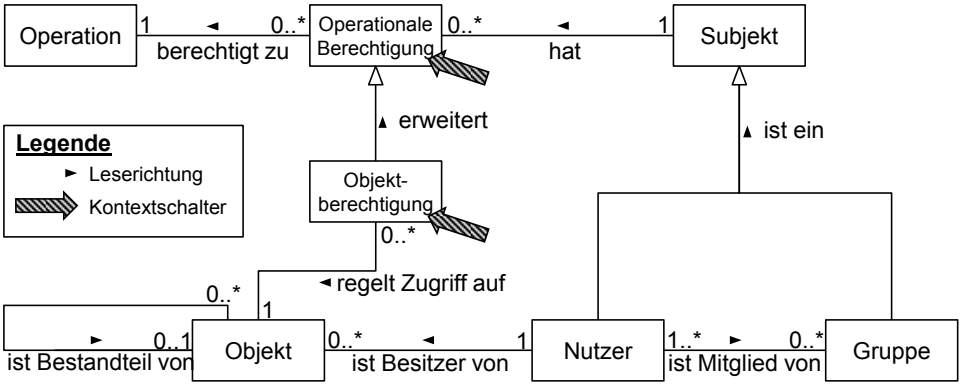

Abb. 3: Allgemeines kontextabhängiges Zugriffsmodell (DAC-Modell)

Folgende Kontextparameter haben sich als besonders geeignet für diese Kontextschalter herausgestellt:

- Ortung des Mobilgeräts
  Die Ortung des Endnutzers kann etwa über GPS (Eigenortung des mobilen Computers) oder die Auswertung der verwendeten IP-Adresse (als Fremdortung durch das Backend) vorgenommen werden. Es kann so z.B. der mobile Zugriff von im Ausland befindlichen Mitarbeitern auf alle Datensätze vom Typ „Patentanmeldung“ verhindert werden.
- Art der Verbindung
  Unterschiedliche Verbindungsarten können verschiedenen Risikoklassen zugeordnet werden und für die Zugriffsentscheidung herangezogen werden. So können z.B. im unternehmenseigenen WLAN Zugriffe auf Daten erlaubt werden, welche an öffentlichen HotSpots verboten sind.
- (Orts-)Zeit
  Für die Zeit als Kontextparameter sind auch die verschiedenen Zeitzonen zu berücksichtigen. Mit diesem Kontextparameter kann etwa verhindert werden, dass ein Nutzer sensible Geschäftsdokumente außerhalb der üblichen Geschäftszeiten ändert. Da für die Bestimmung der Zeitzone auf die Ortung zurückgegriffen werden muss, ist ein Kontextparameter ein Input-Wert für die Ermittlung eines anderen Kontextparameters.
- Termine
  Die Termine eines Nutzers können über seinen vom System verwalteten Terminkalender ausgewertet werden. So kann beispielsweise verhindert werden, dass ein im Urlaub befindlicher Mitarbeiter Informationen verändert.

4.3 Fälschung von Kontextinformation

Zugriffsrechte, welche auf Kontextinformationen beruhen, haben das Problem, dass diese manipuliert werden können. Deshalb sollte eine Zugriffskontrolle niemals alleine darauf beruhen. Die kontextsensitive Zugriffskontrolle dient in diesem Ansatz in erster Linie nicht dem Schutz gegen Angreifer sondern der Unterstützung des Nutzers bei der Einhaltung von Sicherheitsrichtlinien des Unternehmens. Hier ist davon auszugehen, dass die meisten Mitarbeiter sich nicht absichtlich über diese Sicherheitsrichtlinien hinwegsetzen wollen. Gegen unabsichtliche Verletzungen der Sicherheitsrichtlinien sind sie wirksam. Ebenso erfordert die Manipulation der Kontextinformation gezielte Maßnahmen und damit verbunden entsprechende Kenntnisse und Aufwand. Somit kann z.B. ein nur kurzzeitig unbeobachtet gelassenes Gerät nicht zur unbefugten Informationsgewinnung verwendet werden. Ein Beispiel hierfür ist der Besuch eines unabhängigen Beraters beim Kunden A. Wird jetzt das mobile Gerät nach erfolgreicher Authentifizierung durch den Berater kurzzeitig unbeobachtet gelassen, kann kein Mitarbeiter von Kunde A einfach Informationen über den Konkurrenten B abrufen, welcher ebenfalls beraten wird. Dies setzt natürlich voraus, dass entsprechende Regeln vorher festgelegt worden sind.

Eine Manipulation ganz auszuschließen oder auch wesentlich zu erschweren erfordert jedoch einigen Aufwand. Diese Maßnahmen wurden im Rahmen des Projektes nicht weiter vertieft. Das Risiko kann reduziert werden, indem Kontextinformationen aus vertrauenswürdigen Quellen abgerufen oder verifiziert werden, welche nicht durch den Nutzer manipuliert werden können. Beispiele dafür sind: Die aktuelle Zeit wird über einen Zeitserver abgefragt; die Art der Verbindung wird über die IP-Adresse verifiziert; die Termine werden aus dem zentralen Unternehmenssystem

abgerufen. Maßnahmen gegen die Manipulation der Ortungsinformation finden sich z.B. im „Überblick über Ansätze zur Vermeidung der Manipulation von Ortungsverfahren“ [Decker 2009].

5 Lokaler Client-Proxy-Dienst

Zentrale Komponente im mobilen Gerät des Nutzers ist der Client-Proxy-Dienst. Nahezu alle mobilen Geräte bieten die Möglichkeit, einen Proxy zu konfigurieren. Damit ist es möglich, beliebige lokale Anwendungen mit einem Zugriffsschutz über Hardware-Sicherheits-Token und kontextabhängige Zugriffskontrolle zu erweitern. Jegliche Kommunikation des Nutzers mit der Unternehmensanwendung wird über den Proxy umgeleitet. Dieser ermittelt die verfügbaren und benötigten lokalen Kontextinformationen, kommuniziert mit dem Hardwaretoken zur Ermittlung der aktuell gültigen Response und ergänzt die Header der Anfragen des Nutzers an die Unternehmensanwendung um diese Zusatzinformationen. Die Abwicklung des Challenge-Response-Verfahrens wird clientseitig ebenfalls über den Proxy-Dienst realisiert.

Auf Seiten der Unternehmenssoftware können die übermittelten Zusatzinformationen genutzt werden, um die Zugriffsentscheidung zu treffen. Dies kann entweder durch die Anwendung selber (wie im Projekt SumoDacs geschehen) oder durch einen serverseitigen Proxy-Dienst erfolgen. In diesem Fall ist eine Anpassung der Serversoftware evtl. nicht nötig. Sind auf Serverseite Zugriffskontrolle und datenverwaltende Anwendung getrennt, werden im Falle der kontextsensitiven Einschränkungen die Kontextinformationen weitergereicht, da nur die datenverwaltende Anwendung über die Zulässigkeit des Zugriffs entscheiden kann.

6 Praktischer Einsatz

Die entwickelte Lösung konnte im Laborbetrieb erfolgreich getestet werden. Allerdings sind zurzeit noch Hürden im Zusammenhang mit dem praktischen Einsatz vorhanden. Diese hängen vor allem mit dem Zugriff auf den Hardware-Vertrauensanker zusammen. Während der Einsatz mit Desktop-Betriebssystemen funktioniert, sind unter Android Root-Rechte nötig, um auf die Smartcard zugreifen zu können; diese sind in der Regel nicht gegeben. Unter iOS sind noch Fragen mit den Rechten der lokalen Komponente sowie dem Zugriff auf das Sicherheits-Token zu klären.
Die realisierte Lösung basiert deswegen zunächst auf Notebooks mit den Betriebssystemen Windows, Mac OS bzw. Linux und einem Hardwaretoken in Form eines USB-Sticks. Durch geeignete Anpassung des Proxy-Dienstes und der Software zur Ansteuerung der Hardware-Sicherheits-Token (Runtime) können weitere Betriebssysteme und Gerätetypen eingesetzt werden. Eine Anpassung für Android ist in Arbeit, eine Anpassung an weitere Plattformen ist denkbar. Geeignete Bauformen der Sicherheits-Token (z.B. µSD) sind schon verfügbar.

Für den praktischen Einsatz muss außerdem beachtet werden, dass die Verwendung von Kontextinformationen zum Treffen einer Zugriffsentscheidung nur bedingt zu einem höheren Sicherheitsniveau beitragen kann, da die Erhebung dieser Kontextinformationen teilweise leicht manipuliert werden kann (Kontext-Spoofing). Sie dient daher eher der Verhinderung von unbeabsichtigten Fehlern des berechtigten Nutzers oder vor unbedarfteren Angreifern bzw. einem kurzzeitigen Schutz.

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS09035C gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Literatur

[CAS 2013] *CAS Software AG:* CAS PIA - CRM-System in der CRM Cloud - Webbasiertes Mobile-CRM, 2013; *http://www.cas-pia.de*, Abgerufen am 31.10.2013.
[Decker 2009] *Decker, M.:* Ein Überblick über Ansätze zur Vermeidung der Manipulation von Ortungsverfahren. In: Proceedings zur 4. Konferenz "Mobile und ubiquitäre Informationssysteme" (MMS 2009), Münster, Köllen Druck+Verlag GmbH, 2009, S. 53-66.
[Dey 2001] *Dey, A. K.:* Understanding and Using Context In: Personal and Ubiquitous Computing Journal, vol. 5, no. 1, 2001, S. 4-7.
[Ferrari 2010] *Ferrari, E.:* Access Control in Data Management Systems, Synthesis Lectures on Data Management, 2010, Vol. 2, No. 1, Morgan & Claypool Publishers, San Rafael, Kalifornien.
[Miller 1985] *Miller, V. S.*: Use of Elliptic Curves in Cryptography. In: Williams, H. C. (Ed.): Advances in Cryptology - CRYPTO '85, Lecture notes in computer science 218, Springer, Berlin 1986, S. 417-426.
[NSA 2009] *National Security Agency, USA:* The Case for Elliptic Curve Cryptography, 2009; *http://www.nsa.gov/business/programs/elliptic_curve.shtml,* Abgerufen am 31.10.2013.
[WIBU 2011] *WIBU-SYSTEMS AG:* SumoDacs: Secure Mobile Data Access, 2011; *http://www.sumodacs.de,* Abgerufen am 31.10.2013.
[WIBU 2013] *WIBU-SYSTEMS AG:* Ermöglichen Sie eine online Aktivierung Ihrer Software mit der Wibu-Systems CodeMeter-Technologie, 2011; *http://www.wibu.com/de/online-software-aktivierung.html,* Abgerufen am 31.10.2013.
[Yang et al. 1999] *Yang, W.-H. und Shieh, S.-P.:* Password Authentication Schemes with Smart Cards. In: Computer & Security. 1999, Bd. 18, 8, S. 727-733.