Cloud-Sicherheit ohne vertrauenswürdige Administration

Ulrich Greveler[1] · Benjamin Justus[1] · Dennis Löhr[1]

Fachhochschule Münster[1]
Labor für IT-Sicherheit
48565 Steinfurt
{greveler | benjamin.justus | loehr}@fh-muenster.de

Abstract

Der Beitrag beschreibt eine Systemarchitektur für die sichere Speicherung von Daten innerhalb einer Cloud-Applikation. Die Besonderheit stellt dabei ein Trusted-Computing-basierter Mechanismus dar, der sicherstellt, dass weder lokale Systemverwalter noch beim Betreiber der Infrastruktur tätige Cloud-Administratoren einen Zugriff auf die unverschlüsselten Datenbankinhalte vornehmen können. Der erlaubte nutzerseitige Zugriff erfolgt allein über eine *Rule Engine*, die eine maschinenlesbare Rechtebeschreibung (XACML) auswertet. Eine prototypische Implementierung kann der Projektwebseite entnommen werden.

1 Einführung

Das Konzept *Cloud Computing* dominiert seit einigen Jahren die Diskussion um mögliche Kostenersparnisse bei der Auslagerung von IT-Infrastrukturen an Betreiber großer Rechenzentren.

Cloud Computing ermöglicht, abstrahierte und dynamisch skalierbare IT-Dienstleistungen wie Rechenleistung, Speicher, Entwicklungsumgebungen, und komplexe Arbeitsumgebungen von einem zentralen Anbieter netzbasiert und bedarfsabhängig zu beziehen und die Nutzung kostengünstig nach tatsächlichem Aufwand abzurechnen.

In diesem Beitrag konzentrieren wir uns auf das Cloud-Angebot *Software as a Service*, das Zugriff auf auf die in die Cloud ausgelagerte Software über einen Webbrowser realisiert. IT-Administration, Releasemanagement und Wartungsaufgaben werden dabei zentral in der Cloud ausgeführt. Interne Applikationen eines Unternehmens können dabei Finanzbuchhaltung, Administration von Kundenkonten, Bestellabwicklung und Forderungsmanagement sein.

Zu Sicherheitsanforderungen und -betrachtungen von Cloud-basierten Applikationen existieren bereits eine Vielzahl von technischen, teilweise prototypischen, Lösungen und wissenschaftlichen Publikationen. Für die Anbieter der Infrastruktur wurden beispielsweise vom BSI *Mindestsicherheitsanforderungen an Cloud-Computing-Anbieter*[3] entworfen, die u. a. Rechenzentrumsicherheit, Netzsicherheit, Speichervirtualisierung und Schlüsselmanagement betrachten. Zudem wurden Architekturen vorgeschlagen, die eine Speicherung von sensiblen Daten homomorph verschlüsselt[6] vorsehen, so dass zwar eine Verarbeitung der Daten innerhalb der Cloudapplikation möglich ist, ein Zugriff auf die Klardaten seitens der Infrastrukturanbieter

P. Schartner · J. Taeger (Hrsg.) · D•A•CH Security 2011 · syssec (2011) 1-7.

jedoch ausgeschlossen werden kann.

Kernidee vieler Cloud-Sicherheitsarchitekturen ist der Schutz der Applikation bzw. der in der Cloud gespeicherten Daten vor dem Infrastrukturanbieter und vor weiteren Nutzern derselben Infrastruktur, die auf gemeinsame Ressourcen, z. B. Datenbankmanagementsysteme zugreifen. Betrachtete Schwachstellen stellen dabei neben den potentiell nicht vertrauenswürdigen Administratoren der Cloudinfrastruktur mangelnde Isolierung und Kapselung der genutzten Ressourcen und Seitenkanäle zwischen den Cloud-Mandanten dar.

Wir stellen in diesem Beitrag eine Trusted-Computing-basierte Architektur vor, die über den Schutz der Daten vor dem externen Administrator hinaus geht und auch internen Administratoren (d. h. Administratoren des Unternehmens, das seine Applikation in die Cloud auslagert) einen nicht vorgesehen Zugriff auf die in der Cloud gespeicherten Daten wirksam verwehrt.

2 Hintergrund

Für die Architektur der Plattform ausschlaggebend ist der Stand der Technik in Bezug auf vertrauenswürdige Hardware und Trusted Computing. 2003 wurde von führenden IT-Unternehmen die Trusted Computing Group (TCG) gegründet, in der die beteiligten Unternehmen ihre Sicherheitsinitiativen in Bezug auf Trusted Computing koordinieren. Den Kern der Arbeit der TCG bildet die Spezifikation eines Moduls, auf dem das gesamte Sicherheitskonzept aufbaut: das Trusted Platform Module (TPM). Das TPM ist ein passiver Chip, der einen Mikrokontroller enthält und fest mit dem Mainboard oder dem Prozessor verbunden ist. Es ist von seiner Architektur her mit einer Prozessorchipkarte vergleichbar. Wesentliche Funktionen des TPM sind die Bereitstellung eines speziellen Schlüssels, mit dem die Plattform von Dritten als vertrauenswürdig erkannt werden kann, und die sichere Erkennung einer als vertrauenswürdig angenommenen Systemkonfiguration.

Rechtebeschreibungssprachen wie XACML dienen der Kodierung von Rechtebeschreibungen in maschinenlesbarer Form. Unter Rechtebeschreibung (Rights Expression) wird im hier betrachteten Umfeld eine formale Beschreibung verstanden, die ausdrückt, dass einem bestimmten Nutzer (bzw. einer Rolle) ein Recht gewährt oder entzogen wird, unter gewissen Bedingungen eindeutig beschriebene Datenfelder auf eine festgelegte Art und Weise zu nutzen. Die Rechtebeschreibung stellt daher ein formalisiertes Einzelrecht, d. h. eine Zuordnung dieser (max.) fünf Objekte untereinander dar.

Datenschutzfördernde Technologien (Privacy-Enhancing Technologies, PET) realisieren Datenschutz durch technische und kryptographische Verfahren. Das Datenschutz-Ziel wird dabei in die Technik bzw. in die technisch realisierten Protokolle und Verfahren integriert.

Die hier vorgestellte prototypische Lösung [12] ist während der Ausführung des Projektes DaPriM (Data Privacy Management) [1] entstanden. Das BMBF-geförderte Vorhaben umfasst die Entwicklung (d. h. Spezifikation, Dokumentation, Publikation) und prototypische Realisierung einer Technologie zur Speicherung sensibler (i. a. personenbezogener) Daten. Kerngedanke ist dabei, eine Datenspeicherung innerhalb eines abgeschlossenen Systems vorzunehmen, das Operationen auf den Daten vornehmen kann (z. B. Datenabgleich, statistische Auswertung), die gespeicherten Rohdaten aber weder unberechtigten Nutzern noch Administratoren preisgibt. Ein zentrales Element dieser Datenschutz-fördernden Technologie ist der *irreversible Verschluss* [7], der das aus dem Datenschutzrecht bekannte Prinzip der *Datensparsamkeit und Datenvermeidung* technisch erweitert: Daten können unter Nutzung

vertrauenswürdiger Hardware in einer Weise gespeichert werden, die zwar eine Verarbeitung für bestimmte Zwecke erlaubt, ein Abrufen des Datenbestandes jedoch zuverlässig verhindert.

2.1 Verwandte Arbeiten

Maheshwari et al. [8] beschreiben, wie größere Datenmengen (z. B. Festplatten) in unsicherer Umgebung vertrauenswürdig gespeichert werden können, wobei ein kleiner als sicher vorausgesetzter Schlüsselspeicher benötigt wird. Eine praktische Anwendung der Architektur ist Microsoft Bitlocker [4], das das TPM als Schlüsselspeicher verwendet. Das von uns vorgestellte System ist Linux-basiert und verwendet die *TPM-sealing*-Funktion mit einem Key-file für *dm-crypt* mit LUKS[5]. Es wurden zudem Techniken auf Basis von manipulationssicheren Hardware-Token vorgeschlagen [11]. Diese können auf einen Cloud-Service-Anbieter angewandt werden, dem eine Trusted-Computing-Platform zugrunde liegt.

Ein Projekt von Pearson et al. [10, 9] kombiniert einige der genannten Ansätze über die Erstellung eines *Privacy Manager* und einer zugeordneten Rolle, die durch das TPM repräsentiert wird. Das *PrimeLife* Projekt [2] hat ein System entwickelt, das die *PRIME*-Datenschutztechnologien mit XACML-Zugriffskontrollen kombiniert. Alle genannten Systeme fokussieren jedoch nicht den Administrator des vertrauenswürdigen Systems als potentiellen Angreifer.

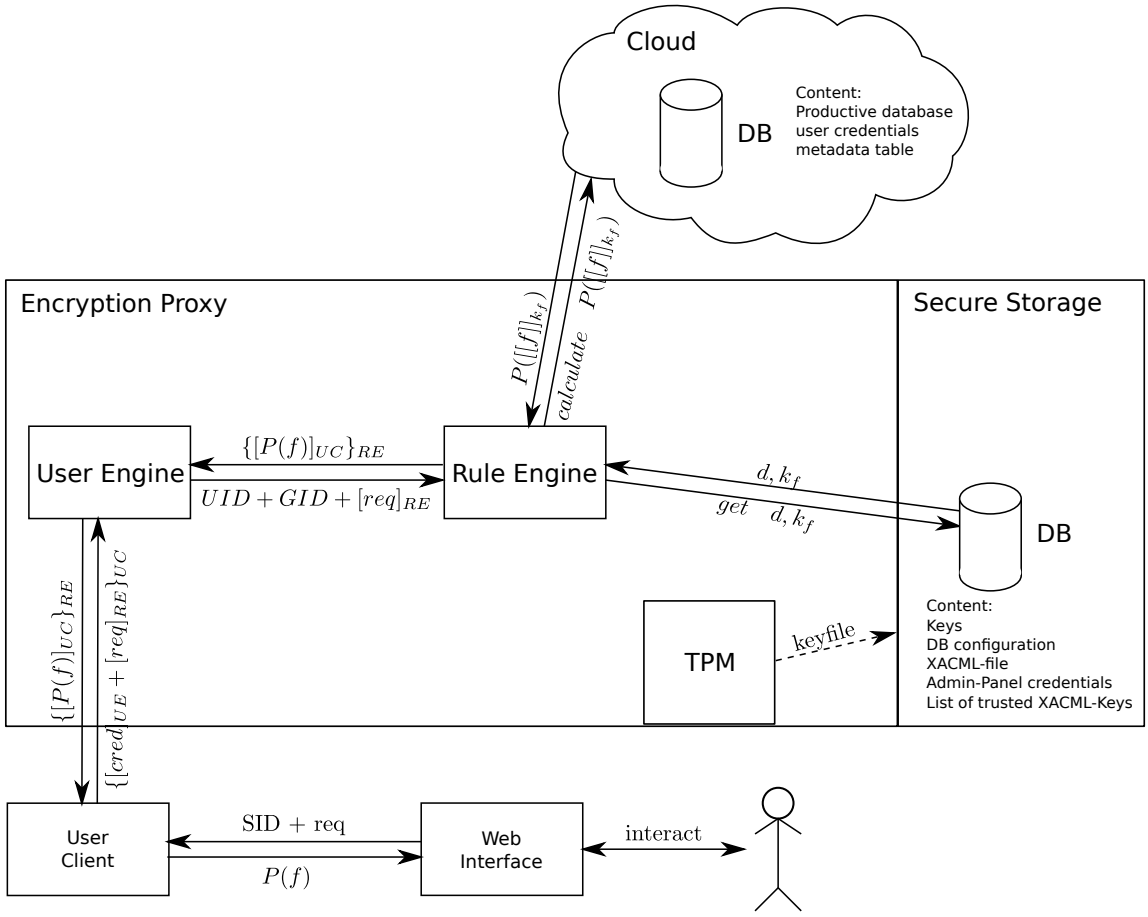

Abb. 1: Encryption Proxy

3 Architektur der Lösung

Die Produktivdaten der Applikation sind in die Cloud ausgelagert und liegen dort verschlüsselt vor. Der nutzerseitige Zugriff auf die Daten erfolgt über einen der *Encryption Proxys*, die eine transparente Ver- und Entschlüsselung der Datenbankinhalte vornehmen. Der Proxy kann beliebig gewählt bzw. über einen *Load Balancer* zugewiesen werden, um eine Lastverteilung zu ermöglichen.

Eine grafische Übersicht der Architektur wird in der Abb. 1 gegeben.

Innerhalb der Produktivdatenbank wird eine Tabelle mit Metadaten verwaltet, die Informationen über die Transaktionen der User speichert. So können beispielsweise Zähler geführt werden, die eine über die Rechtebeschreibung definierte maximale Anzahl von Zugriffen eines bestimmten Nutzers überwachen. Diese Tabelle wird signiert gespeichert; Signaturen können nur von solchen Proxys über den TPM-Chip erzeugt werden, die in einer als sicher angenommenen Systemkonfiguration gestartet wurden (*TPM Quote Function*). Ein Proxy liest die von einem *XML Editor* (Rolle ist unabhängig vom lokalen Systemverwalter) signierte Rechtebeschreibung (eine Beispieldatei wird im Listing 3 gezeigt) und lässt mithilfe einer *Rule Engine* genau die Zugriffe auf die Cloud-Applikation zu, die gemäß der Rechtedefinition erlaubt sind.

Ist der Proxy in seiner sicheren Konfiguration gebootet, hat der Systemverwalter keinen Zugriff über eine Login-Shell o. ä. Bootet er das System für die Vornahme administrativer Eingriffe, ist kein Zugriff auf die Datenbankinhalte der Cloud möglich, da die TPM-basierte Freigabe auf Entschlüsselungsschlüssel im Wartungszustand des Systems nicht vorhanden ist. Die Neudefinition einer sicheren Systemkonfiguration (z. B. nach einem Betriebssystemupdate) ist nur im 4-Augen-Prinzip zwischen lokalem Administrator und *XML-Editor* möglich. Eine Übersicht der unterschiedenen Rollen des Systems ist in Tabelle 1 gegeben.

Die prototypische Implementierung der von uns vorgestellten Architektur kann vom Projekt-Webserver[1] heruntergeladen werden.

Rolle	Bezeichner	Rechte
System-Administrator	A	Backup: verschlüsselte Daten & Einrichten der *Encryption Proxys*
XACML editor	$E_1, E_2, ..., E_n$	entwickeln und aktivieren neuer Zugriffsregeln
Developer	D	Neue *Requests* erstellen
Employee/Worker	$W_1, W_2, ..., W_m$	Zugriff auf beschränkte Anzahl von Datensätzen

Tab. 1: Übersicht: Rollen

[1] www.daprim.de – Data Privacy Management: Entwicklung einer datenschutzfördernde Technologie auf Basis digitaler Rechteverwaltung (DaPriM) – Förderkennzeichen 17076X10 (Förderprogramm FHprofUnt, Finanzierungsrunde 2010)

Listing 1: Struktur eines XACML-Files mit zwei Signaturen.

```
<?xml version="1.0" encoding="UTF-8">
<Policy xmlns:... PolicyId="paper"
RuleCombiningAlgId="urn:oasis:...">
 <Description>
  This is an example XACML-file
 </Description>
 <Target>
  <Subjects> <AnySubject/> </Subjects>
  <Resources> <AnyResource/> </Resources>
  <Actions> <AnyAction/> </Actions>
 <Target>
 ...
 <Rule RuleId="..." Effect="Permit">
  <Description>
   This is an example rule
  </Description>
  <Target>
   <Subjects> <AnySubject/> </Subjects>
   <Resources> <AnyResource/> </Resources>
   <Actions> <AnyAction/> </Actions>
  </Target>
  <Condition>
   ...
  </Condition>
 </Rule>
 ...
</Policy>
<Signature id="sig1">
 <SignedInfo>...</SignedInfo>
 <SignatureValue>...<SignatureValue>
 <KeyInfo>...<KeyInfo>
</Signature>
<Signature id="sig2">
 <SignedInfo>...</SignedInfo>
 <SignatureValue>...<SignatureValue>
 <KeyInfo>...<KeyInfo>
</Signature>
```

Da der nutzerseitige Zugriff hier allein über die *Rule Engine* erfolgen kann, ist eine wirksame Beschränkung der legitimen Nutzer auf (wenige) klar definierte Zugriffsmöglichkeiten umsetzbar. So kann insbesondere verhindert werden, dass ein Mitarbeiter des Unternehmens, das seine Kundendaten in die Cloud ausgelagert hat, eine Kopie des Stammdatenbestandes vornehmen und außerhalb der Cloud bearbeiten kann. Selbst wenn dieser Mitarbeiter Zugriff auf alle Stammdaten benötigt (wie z. B. ein Mitarbeiter des Inbound-Callcenters), kann wirksam unterbunden werden, dass der Mitarbeiter eine bestimmte Anzahl von Leseoperationen, die über seine notwendigen Privilegien hinausgehen, ausführt. Vorsätzlich wie auch fahrlässig erzeugte Datenbankkopien, die bei den sogenannten Datenpannen, über die in der jüngeren Vergangenheit berichtet wurde, eine Rolle spielen, sind damit nicht erzeugbar.

4 Diskussion und Fazit

Die sensiblen Daten, die innerhalb der Cloud gespeichert werden, können im von uns vorgeschlagenen System nur gemeinsam mit einer *usage policy* abgelegt werden. Der Zugriff erfolgt dann gemäß einer maschinenlesbaren Rechtebeschreibung, die im Zusammenwirken mit Trusted Computing sicherstellt, dass die Nutzer nur im für sie vorgesehen Rahmen die Applikation verwenden (insbesondere keine Kopie des Gesamtdatenbestandes vornehmen können) und kein unkontrollierter Zugriff seitens lokaler oder externer Administratoren erfolgt.

Zukünftige Herausforderungen liegen in der effizienten und zuverlässigen Erstellung der Rechtebeschreibung. Zwar erlaubt die umfangreiche Syntax heute verfügbarer Rechtebeschreibungssprachen komplexe und fein-granulare Rechtedefinitionen, es ist jedoch keine triviale Aufgabe, sicherzustellen, dass diese Definition genau die Politik abbildet, die für die Applikation festgelegt wurde. Eine potentiell fehlerhafte Rechtedefinition, die einem Nutzer großzügig Privilegien einräumt, stellt eine wesentliche Schwachstelle der Architektur dar.

Die aufgrund der Systemarchitektur vorgenommene Aufspaltung der privilegierten Benutzerrollen in Administratoren und Rechteeditoren lässt eine Lösung zu, die weder (lokalen) Administratoren noch Cloud-Administratoren, die beim Infrastrukturanbieter tätig sind, einen Zugriff auf die unverschlüsselten Datenbankinhalte einräumt. Wir halten diese Eigenschaft für zukunftsweisend, da das implizit angenommene Vertrauen in die Adminstration bei ähnlichen Lösungen eine – angesichts der Sensibilität vieler personenbezogener Daten – kaum noch akzeptable Grundvoraussetzung für eine Cloud-basierte oder auch lokale Datenspeicherung in größerem Rahmen darstellt.

References

[1] www.daprim.de (Webauftritt) – Data Privacy Management: Entwicklung einer datenschutzfördernde Technologie auf Basis digitaler Rechteverwaltung (DaPriM). Fachhochschule Münster, Labor für IT-Sicherheit.

[2] C.A. Ardagna, S. De Capitani di Vimercati, S. Paraboschi, E. Pedrini, and P. Samarati. An XACML-based privacy-centered access control system. In *Proceedings of the first ACM workshop on Information security governance*, pages 49–58, New York, NY, USA, 2009. ACM.

[3] BSI. *BSI-Mindestsicherheitsanforderungen an Cloud-Computing-Anbieter, Stand 27.09.2010, ENTWURF*. BSI Bonn, 2010.

[4] N. Ferguson. AES-CBC+ Elephant diffuser A Disk Encryption Algorithm for Windows Vista. 2006.

[5] C. Fruhwirth. LUKS On-Disk Format Specification Version 1.1. 2005.

[6] Craig Gentry. Fully homomorphic encryption using ideal lattices. *41st ACM Symposium on Theory of Computing (STOC)*, 2009.

[7] Ulrich Greveler. Irreversibler Verschluss: DRM-basierter Datenschutz. *in Patrick Horster (Hrsg.), D.A.CH Security '09, syssec*, 2009.

[8] Umesh Maheshwari, Radek Vingralek, and William Shapiro. How to build a trusted database system on untrusted storage. In *Proceedings of the 4th USENIX Symposium on Operating System Design and Implementation*, Berkeley, CA, USA, 2000.

[9] Miranda Mowbray, Siani Pearson, and Yun Shen. Enhancing privacy in cloud computing via policy-based obfuscation. pages 1–25. Springer Berlin / Heidelberg, 2010.

[10] S. Pearson, Y. Shen, and M. Mowbray. A privacy manager for cloud computing. In *Cloud Computing*, Lecture Notes in Computer Science, pages 90–106. Springer Berlin / Heidelberg, 2009.

[11] Ahmad-Reza Sadeghi, Thomas Schneider, and Marcel Winandy. Token-based cloud computing. In Alessandro Acquisti, Sean Smith, and Ahmad-Reza Sadeghi, editors, *Trust*

and Trustworthy Computing, volume 6101 of *Lecture Notes in Computer Science*, pages 417–429. Springer Berlin / Heidelberg, 2010.

[12] Benjamin Justus Ulrich Greveler and Dennis Löhr. A Privacy Preserving System for Cloud Computing. *The 2011 International Workshop on Survivable Large-Scale Information Systems, Cyprus. IEEE Computer Society Press*, 2011.