Sichere Datenhaltung in der Cloud via Handy

Tuba Yapinti

Abschlussvortrag der Bachelorarbeit

Betreuer: Prof. Reinhardt, Dr. Bernd Borchert

Gliederung

1. Motivation
   - Gründe für die Entwicklung
   - Ideen für die Entwicklung
2. Einleitung
   - Cloud-Computing
   - AES und MD5
   - eKaay
3. Entwicklung
   - Login
   - Datenhaltung
4. Fazit

1. Motivation

- Kunden Daten von größeren Unternehmen wurden von Hackern geklaut!!

  - *70 Millionen Kundendaten wie z.B. Kreditkartendaten Passwörter etc. wurde dem Unternehmen Sony geklaut(April 2011)*

  - *Die Anzahl Cloud Nutzer steigen an, z.B. Aktionen wie Spacerace von dropbox locken immer mehr Nutzer an. Oktober 2012 bis November 2012 sind 47536 Nutzer dazu gekommen.*

1. Motivation

- Gründe
  - *Zunahme der Nutzung von Dienstleistungen im Internet*
  - *Nutzer speichern wichtige Daten auch im Internet*
  - *Sicherheit der Daten ist nicht ausreichend gewährleistet(z.B. Daten sind nicht verschlüsselt abgespeichert)*
  - *Sicherheitslücken im Login-Verfahren mit Passwort*

1. *Wie kann man Nutzer Passwort besser schützen?*

2. *Wie schützt man die Daten des Nutzers?*

3. *Gibt es für den Nutzer ein alternatives Login-Verfahren?*

1. Motivation

- Ideen
  - *Alternative Login Verfahren **eKaay***
  - *Passwort mit Salz erweitern und gehasht auf dem Server speichern.(**MD5)***
  - *Dateien nur mit Chiffretext auf dem Server speichern mit (**AES)***

Einführung

2. Einführung

- Cloud-Computing

  - *Besteht aus drei Bausteinen:*

    - *Software as a Service (SaaS), Anwendungen werden dem Nutzer zu Verfügung gestellt.*
    - *Platform as a Service (PaaS), Frameworks und andere Anwendungen zur Webentwicklung.*
    - *Infrastuctur as a Sevice (IaaS), Netzwerkdienste oder Server*

2. Einführung

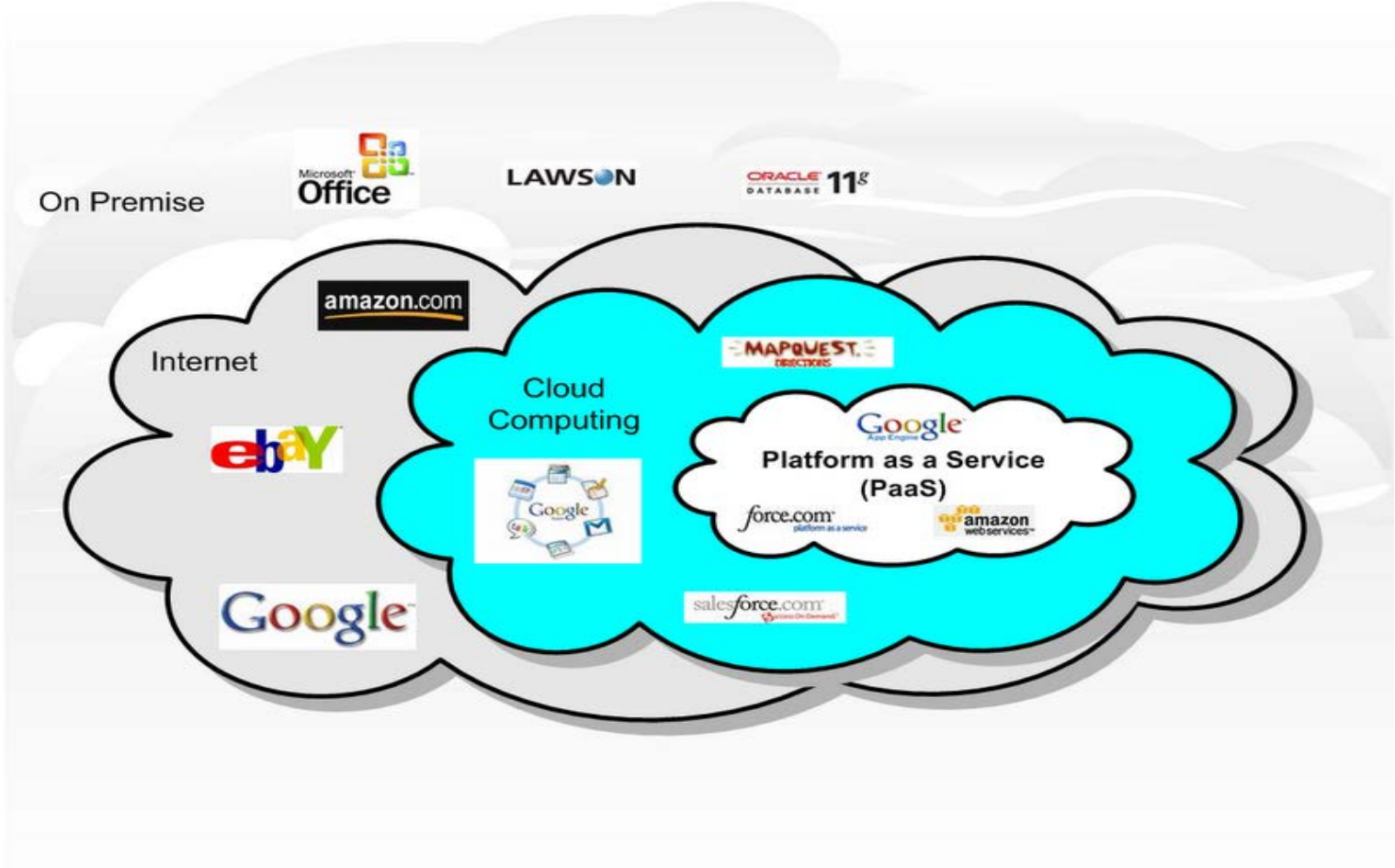

Bild 1: Beispiele für Cloud Anbieter [9]

2. Einführung

- Advanced Encryption Standart-AES

  - *Symmetrisches Verschlüsselungsverfahren, d.h. für Verschlüsselung und Entschlüsselung wird der selbe Schlüssel verwendet.*

  - *Der Klartext wird in Datenblöcke von 4x4-Bit umgewandelt und der Benutzerschlüssel(Passwort) wird in ein Chiffreblock geschrieben.*

  - *Algorithmus durchläuft mehrere Runden in den immer unterschiedliche Chiffren berechnet werden.*

  - *Um Chiffretext zu entschlüsseln wird der Algorithmus rückwärts angewendet.*

2. Einführung

- *Je Runde werden vier Operationen durchgeführt.*

  - *AddRoundKey:*
    Datenblock wird mit Chiffreblock XOR-verknüpft

  - *SubBytes:*
    Einträge des Datenblock werden mit den Werten aus der Substitutionstablle ersetzt.

  - *ShiftRows:*
    Zeilen des Datenblocks werden nach links verschoben
  - *Mixcolums:*
    Spalten des Datenblocks werden mit einer komplexen mathematischem Verfahren geändert.

2. Einführung

- Message-Digest Algorithmus-*MD5*
  - *Der Algorithmus erzeugt aus einer Eingabe einen 128-Bit Hashwert*
  - *Für die Umwandlung wird die Eingabe so codiert das ein 512-Bit großer Block entsteht*
  - *Datenblock wird in kleinere Teile aufgeteilt*
  - *Datenteilblöcke werden mit einer Komprimierungsfunktion bearbeitet*

2. Einführung

- *eKaay*

Bild 2: eKaay Login-Ablauf [5]

2. Einführung

- eKaay
  - *Server erzeugt einen 2D-Code als Challenge*
  - *Nutzer nimmt die Challenge mit seinem Handy mit Hilfe der eKaay-App auf*
  - *Handy bearbeitet die Challenge und schickt einen Response an den Server*
  - *Ist der vom Handy gesendete Schlüssel dem Server bekannt, so wird eine Verbindung mit dem Client aufgebaut und der Nutzer wird mit Name und Passwort eingeloggt.*

Entwicklung

3. Entwicklung-Login

1. *Nutzer muss mit seinem Handy den 2D-Code scannen, ist der Nutzer registriert wird er auf die zweite Seite weitergeleitet.*

2. *Nutzer gibt sein Passwort ein, dass zu Ver -und Entschlüsselung des Klartextes benötigt wird.*

→ *Ist das Login erfolgreich beendet, so kann der Nutzer seine Datei bearbeiten.*

3. Entwicklung-Login

- Probleme
  - *Der Schlüssel muss geschützt werden, denn kennt der Hacker den Schlüssel kann er die Dateninhalte einsehen und ändern.*

Lösung:

1. *Passwort wird gehasht*
2. *Passwort wird mit einem Salt erweitert*
3. *Passwort nur gehasht auf dem Server gespeichert*
4. *Passwort wird vor dem Senden der Formdaten gehasht*

3. Entwicklung-Login

- Effizienz der Lösung

Passwort wird mit MD5 gehasht

ABER:

- *Wählt Nutzer ein kurzes Passwort, Gefahr durch Brute-Force-Angriff*
- *Wählt der Nutzer einen simples Passwort, ist das der Nutzer gegen Regenbogentabellen ungeschützt*

3. Entwicklung-Login

- Effizienz der Lösung

Passwort wird mit einem Salt erweitert

Passwort wird mit dem Nutzernamen erweitert und dann mit MD5 gehasht.

Vorteil:

- *Brute-Force viel zu aufwendig und benötigt enorme Rechenzeit*
- *Regenbogentabellen für Lange Passwörter zu rechenaufwendig nahe zu unmöglich.*

3. Entwicklung-Login

- Effizienz der Lösung

Passwort nur gehasht auf dem Server gespeichert

Vorteil:

- *Passwort wegen dem Salt schwer zu regenerieren, gehashter Passwort ist für Hacker nutzlos*
- *Hacker kann sich mit gehashtem Passwort nicht einloggen*

ABER:

- *Hacker kann Request und Response zwischen Server und Client das ungehashte Passwort klauen*

3. Entwicklung-Login

- Effizienz der Lösung

> *Passwort wird vor dem Senden der Formdaten gehasht*

Das Passwort wird mit einer JavaScript Funktion noch am Client gehasht.

Vorteil:

- *Passwort wird vom Client nie im Original an den Server geschickt*
- *Server vergleicht nur gehashte Werte miteinander*
- *Hacker kann während Client Server Kommunikation kein Original klauen*

3. Entwicklung-Datenhaltung

1. *Datei des Nutzers ist auf Server gespeichert und immer abrufbar*

2. *Nutzer kann neue Texte schreiben oder alten Text bearbeiten*

3. Entwicklung-Datenhaltung

- Probleme
  - *Hacker können Dateninhalt vom Server klauen*
  - *Beim Abruf und Speichern des Klartextes können die Inhalte angefangen werden*

Lösung:

1. *Dateninhalte nur verschlüsselt auf dem Server speichern*
2. *Ver- und Entschlüsseln auf dem Client ausführen*

3. Entwicklung-Datenhaltung

- Effizienz der Lösung

> *Dateninhalte nur verschlüsselt auf dem Server speichern*

Vorteil:

- *Ohne das Passwort kann der Angreifer mit dem Chiffretext nichts anrichten*

ABER:

- *Hacker kann während Client Server Kommunikation den Klartext lesen*

3. Entwicklung-Datenhaltung

- Effizienz der Lösung

> *Ver- und Entschlüsseln auf dem Client ausführen*

Client sendet nur verschlüsselte Daten an den Server und Daten vom Server werden nur über JavaScript Funktionen mit dem AES-Algorithmus ver- und entschlüsselt

Vorteil:

- *Datenfluss zwischen Server und Client ist für Angreifer nutzlos denn nur Chiffretext wird gesendet, so kann der Angreifer ohne Passwort nichts mit dem Chiffretext anrichten*

Fazit

4. Fazit

- *eKaay*

  - *Hacker möchte sich im Account vom Nutzer einloggen, hat aber das Handy nicht.*

    „Ist Token dem Server unbekannt oder versucht der Nutzer sich ohne Handy auf die zweite Seite zu gelangen, sieht der Nutzer den geheimen Inhalt nicht“

  - *Hacker versucht die aktuelle Session zu klauen.*

    „Token wird immer bei einer Server-Client Kommunikation geprüft ob dieser noch gesetzt ist, falls nicht wird Session abgebrochen. So kann die Session nicht von Dritten geklaut werden“

4. Fazit

- *MD5*
  - *Kennt man das Passwort nicht, so kann man den Inhalt der Dritten Seite nicht sehen, wenn der Hacker das Handy besitzt.*

    „Versucht der Angreifer durch Brute-Force das Passwort zu erraten, wird er an dem Rechenaufwand scheitern, denn Passwort ist zu Lang“

    „Versucht der Hacker das Passwort mit Regenbogentabellen zu regenerieren, scheitert er ebenfalls an dem zu großen Rechenaufwand“

4. Fazit

- *AES*

  - *Daten auf dem Server werden von Hacker geklaut.*

    „Dateninhalt ist verschlüsselt und ohne Passwort kann der Klartext nicht regeneriert werden, so lange der Hacker das Passwort nicht kennt ist dass regenerieren zu rechenaufwendig “

Quellen

[1] Advanced Encryption Standard (AES)
http://www.dpunkt.de/leseproben/3112/Kapitel%208.pdf

[2] Christoph Wille,Passwörter speichern - aber richtig!
http://www.aspheute.com/artikel/20040105.html

[3] Marco Schürmann,Georg Pichler,Cloud-Computing gewinnt an Bedeutung
http://www.beyond-print.de/2012/03/16/2014-cloud-als-zentrum-des-digitalen-lebens/

[4] Wolfgang Herrmann, Technik der Zukunft-Wie der Hype entstand
http://www.pcwelt.de/ratgeber/Wie-der-Hype-entstand-Technik-der-Zukunft-228731.html

[5] Dr. Bernd Borchert} eKaay Implementierung
http://www.ekaay.com/implement/

[6] Dr. Bernd Borchert, eKaay - Sicherheit
http://www.ekaay.com/security/

[7] Alexey Schröder, Passwort, MD5-Hash und Salt
http://acadopus.de/security/passwort-md5-hash-und-salt_2855.html

[8] Nils Reimers, Was ist md5?
http://www.php-einfach.de/wissen_md5.php

[9] Teambox
http://www.cloud-computing-network.com/