Kryptografisches Zugriffskontrollsystem für mobile Endgeräte

Ernst Piller[1] · Andreas Westfeld[2]

[1]Fachhochschule St. Pölten
ernst.piller@fhstp.ac.at

[2]Hochschule für Technik und Wirtschaft Dresden
andreas.westfeld@htw-dresden.de

Zusammenfassung

Bei der Verwendung von mobilen Endgeräten, insbesondere von Smartphones, hat die Speicherung von Daten in einem Cloudspeicher oftmals eine große Bedeutung. Da dabei meist auch die Betreiber der Cloudspeicher einen Zugang zu den Daten haben, ist eine ausreichende Datensicherheit nicht gewährleistet. Im vorliegenden Artikel wird im ersten Kapitel die Forschungsfrage und im zweiten Kapitel die Ausgangslage dargestellt. Im dritten Kapitel erfolgt eine kurze Vorstellung mehrerer am Markt verfügbarer Lösungen für die Datenverschlüsselung bei der Verwendung von Cloudspeichern. Unter Berücksichtigung der Sicherheitsproblematik von Cloudspeichern und den Engpässen von Smartphones in Bezug auf Verarbeitungs- und Übertragungsgeschwindigkeit, Speichervolumen und Datensicherung/Backup (hier im Speziellen von kryptografischen Schlüsseln) wurde im Forschungsprojekt „Smartphone Security“ ein Anforderungskatalog an ein kryptografisches Zugriffskontrollsystem für Smartphones/Tablets erstellt. Daraus wurde für ein neues kryptografisches Zugriffskontrollsystem für Smartphones/Tablets ein Systemkonzept und Schlüsselmanagementsystem erforscht und ein realer Prototyp (App) für Android Plattformen entwickelt, das im vierten Kapitel vorgestellt wird. Dieses System, nachfolgend CASSP genannt, berücksichtigt unter anderem das Rabin Kryptosystem [Bart12, Rabi79], verwendet zur Optimierung der modularen Multiplikation sehr großer Zahlen die Bunimov-Methode [BuSc05, ScBu04] und für die Sicherung der kryptografischen Schlüssel ein (k, n)-Schwellwertschema [RoDe82, Sham79]. Des Weiteren erfordert CASSP keine Neuverschlüsselung bei der Änderung von Zugriffsberechtigungen und entlastet damit die Endgeräte in hohem Maße.

1 Forschungsfrage

Die Forschungsfrage war, welche Anforderungen an ein kryptografisches Zugriffskontrollsystem für Smartphones/Tablets zur Datenspeicherung in Cloudspeichern zu stellen sind, welche heute von bekannten Verfahren und Lösungen erfüllt werden und wie eine neue Lösung, die sich speziell auf Zeiteffizienz am Smartphone/Tablet und Sicherheit orientiert, aussehen kann. Der Artikel behandelt diese Fragestellungen und liefert als Antwort ein neuartiges kryptografisches Zugriffskontrollsystem.

P. Schartner · K. Lemke-Rust · M. Ullmann (Hrsg.) · D•A•CH Security 2015 · syssec (2015) 231-240.

2 Einleitung

Der massive Boom bei Smartphones und Tablets und die verhältnismäßig geringen Kosten bei den Datentarifen führen dazu, dass immer größere Mengen an Daten von Smartphones / Tablets über das Internet verschickt werden. Dabei wird es immer beliebter, Daten in einer Cloud zu speichern und via Cloud mit anderen zu teilen [Mats08]. Die Kosten dafür sind sehr gering (keine Wartungskosten, kostenlos für private Benutzer) und die Verfügbarkeit ist sehr hoch. Es existiert heute schon eine Vielzahl an Cloudspeicher-Anbietern. Dropbox, Google Drive, Box, Microsoft OneDrive (SkyDrive) und Amazon Cloud Drive zählen zu den prominentesten unter ihnen, aber auch weniger bekannte wie z.B. Egnyte, FileSpots oder Livedrive haben nennenswerte Markanteile. Fast alle Anbieter bieten dabei kostenlosen Speicherplatz für Privatperson mit Optionen zum kostenpflichtigen Upgrade und Lösungen für Unternehmen. Dass das Speichervolumen für Privatpersonen kostenlos ist, ist natürlich ein entscheidender Faktor für die Popularität von Cloud-Services und hat zu deren Verbreitung beigetragen. Ein weiterer Faktor ist sicherlich auch die zunehmende Bandbreite für Privatpersonen.

Die zunehmende Veröffentlichung von Sicherheitsschwachstellen lassen bei vielen Benutzern aber Bedenken an die Cloud Anbieter aufkommen, wie z.B. der Fall mit dem Datenklau von vielen Prominenten aus Apple's iCloud aufzeigt [Stutt]. Um die Sicherheit zu erhöhen, verwenden viele Anbieter wie Dropbox Verschlüsselungstechnologien wie SSL/TLS bei der Kommunikation zwischen Client und Server, um einen Man-in-the-middle-Angriff auszuschließen. Eine sichere Datenübertragung alleine reicht leider nicht aus, um zu garantieren, dass die Daten auch sicher in der Cloud abgelegt sind. Dritte, wie z.B. die Administratoren des Cloud Anbieters, können sich meist Zugang zu den Daten beschaffen. Aktuell unterstützt kein größerer Cloud Anbieter das Verwenden eines eigenen geheimen Schlüssels des Benutzers für die Verschlüsselung der Daten auf den Servern [Chu13, Jaeb12].

Folgende Tabelle zeigt eine Übersicht über einige weitere Sicherheitslücken [Chu13] der Cloud-Speicher Anbieter Dropbox, Google Drive und Microsoft OneDrive (SkyDrive) (X bedeutet, es gibt eine Schwachstelle).

Tab. 1: Übersicht über Sicherheitslücken der Cloudspeicher Anbieter

Schwachstelle	Dropbox	Google Drive	Microsoft Skydrive
Nondead URL	X	–	–
Uncertain identities	X	X	X
Unauthorized resharing	X (secret URL sharing)	X (private sharing)	–
Indiscriminate accessing URL	–	X	–
NonHTTPS shortened URL	X	–	X
No privacy on sharing	X	X	–
Sharing of trash files	–	X	–
Fixed URL	–	X	–

Nondead URL: Eine Datei wird auf dem Cloud-Speicher geladen und mittels geheimer URL

freigegeben. Diese Datei ist dann sichtbar für alle, die den Link haben. Wird diese Datei gelöscht und eine neue Datei mit demselben Namen im gleichen Verzeichnis erstellt, ist es möglich, über die "alte" URL Zugriff auf die neue Datei zu erhalten.

Uncertain identities: Bekommen Benutzer den Zugriff auf einer Datei erteilt, sind sie nicht daran gebunden, das Konto des Cloud-Speicher Anbieters zu verwenden, das ursprünglich zum Senden der Einladung referenziert wurde. Es kann auch jedes beliebige andere Konto verwendet werden. Die Zugriffsliste einer Datei kann zu einem späteren Zeitpunkt unter Umständen Konten auflisten, denen nie Zugriff für diese Datei gewährt wurde.

Unauthorized resharing: Das heißt, dass Benutzer, die Zugriff auf eine Datei über eine geheime URL bekommen haben, diese geheime URL beliebig oft weitergeben können, ohne dass Einfluss darauf genommen werden kann.

Indiscriminate accessing URL: Bei Google Drive wird eine Datei immer über dieselbe URL referenziert. Beispielsweise kann in einem Meeting jeder die URL einer Datei sehen, die aus dem Google Drive auf einem Projektor geöffnet wurde. Es kann daher über diese URL auf die Datei zugegriffen werden, wenn sie zufälligerweise über eine geheime URL freigegeben wurde.

NonHTTPS shortened URL: Dropbox und Microsoft OneDrive bieten für Mobilgeräte ein URL-Verkürzungsservice an. Die Kurz-URLs verwenden dabei kein HTTPS. Die URL-Pfade sind dann nicht verschlüsselt und somit nicht gegen Abhörangriffe geschützt.

No privacy on sharing: Jeder Benutzer, der Zugriff auf eine freigegebene Datei hat, kann sehen, wer noch Zugriff auf diese Datei hat.

Sharing of trash files: In Google Drive können Dateien in den Papierkorb geschoben werden, bevor sie endgültig gelöscht werden. Dies passiert automatisch, wenn auf Löschen geklickt wurde. Bei Google Drive kann jedoch auf Dateien im Papierkorb immer noch von allen Benutzern, die ursprünglich Zugriff auf die Datei hatten, zugegriffen werden.

Fixed URL: Bei Google Drive wird auf eine Datei immer über dieselbe URL zugegriffen. Eine öffentlich freigegebene URL kann nicht auf eine geheime URL geändert werden.

3 Ausgangslage

Grundsätzlich sind zwei verschiedene Typen von Verschlüsselungssystemen für Cloud-Speicher verbreitet:

- Container-basierte ohne Schlüsselmanagement und
- Datei-basierte mit Schlüsselmanagement.

Container-basierte Systeme bieten Verschlüsselung nur für komplette Container, also nur für eine ganze Sammlung von Dateien als Ganzes. Ein bekanntes Beispiel für diese Art von Verschlüsselungssystemen ist TrueCrypt [TC14, Sasc12, Geer13]. Mit Hilfe von TrueCrypt kann ein verschlüsseltes Abbild eines Dateisystems erzeugt werden. Das verschlüsselte Abbild kann dann auf einen Cloud-Speicher hochgeladen werden. Andere Benutzer können auf die inkludierten (verschlüsselten) Dateien nur zugreifen, wenn der Datei-Container mittels TrueCrypt entschlüsselt wird, wozu der entsprechende Schlüssel notwendig ist. TrueCrypt selbst bietet allerdings kein Schlüsselmanagement und auch keine Freigabe-Funktionalitäten an, es handelt sich um eine reine Verschlüsselungssoftware. Im Jahr 2014 wurde für TrueCrypt wegen angeblicher Sicherheitslücken der Support eingestellt, das Unternehmen weist auf seiner Website auf BitLocker als Alternative hin [Half14]. Ein weiteres Beispiel für Container-basierte Systeme ist

der Online-Service EncipherIt [Enci14, Sasc12]. Mit der Software von EncipherIt können Dateien und ganze Container verschlüsselt werden, bevor sie auf einen Cloud-Speicher hochgeladen werden. Später kann man diese mit derselben Software wieder entschlüsseln. Auch hier werden keinerlei Schlüsselverwaltungsfunktionen angeboten.

Im Gegensatz zu Container-basierten Verschlüsselungssystemen arbeiten Datei-basierte Systeme auf Dateibasis und bieten sehr oft auch ein integriertes Schlüsselmanagement an. Die meisten derartigen Systeme, die heute verfügbar sind, verfügen außerdem über eine sehr gute Integration in viele Cloud-Speicher Lösungen. Sehr bekannte Produkte sind z.B. Boxcryptor [Boxc14, Sasc12, PaSo13, Geer13], CloudFogger, DigitalQuick oder Viivo.

Beispielsweise verwendet Boxcryptor ein hybrides Verschlüsselungsverfahren mit AES (256 Bit Schlüssellänge), CBC Mode, PKCS7 Padding und zur Schlüsselverteilung RSA-OAEP (4096 Bit Schlüssellänge). Es unterstützt die Cloud-Speicher Dropbox, Google Drive, Box, Microsoft OneDrive und andere Anbieter, die das WebDAV Protokoll verwenden. Des Weiteren ist die Boxcryptor-Software für alle gängigen Plattformen (Windows, OS X, Android, iOS,...) frei verfügbar und das Basispaket ist für private Zwecke gratis. Es gibt aber auch mehrere Pakete, die mehr Funktionen bieten, bzw. die speziell auf Unternehmen abgestimmt sind. Zusätzlich zu der Verschlüsselung des Dateninhalts ist es mit Hilfe von Boxcryptor auch möglich, die Dateinamen am Cloud-Speicher zu verschlüsseln. Außerdem bietet Boxcryptor ein System zur Zugriffssteuerung von Dateien und Ordnern und es ist möglich Firmenrichtlinien (Policies) zu definieren.

4 CASSP

Im Rahmen des Forschungsprojektes „Smartphone Security“, gefördert durch die österreichische Forschungsförderungsgesellschaft (FFG, Programmlinie KIRAS, finanziert durch das Bundesministerium für Verkehr, Innovation und Technologie), entstand das innovative kryptografische Zugriffskontrollsystem CASSP (Cryptographic Access Control Solution for Smart Phones) in Form eines umfangreichen Konzeptes, Schlüsselmanagementsystems und eines realen Prototyps (Apps) für Android Plattformen. Vor den Forschungs- und Entwicklungstätigkeiten von CASSP wurde ein ausführlicher Anforderungskatalog erstellt, der vor allem die Sicherheitsproblematik von Cloudspeichern und Engpässe von Smartphones in Bezug auf Verarbeitungs- und Übertragungsgeschwindigkeit, Speichervolumen und Datensicherung (Backup) – hier im Speziellen von kryptografischen Schlüsseln – berücksichtigt. Bei der Ausarbeitung des darauf folgenden Lösungskonzepts inklusive dem kryptografischen Konzept wurde versucht, alle diese Anforderungen bestmöglich zu erfüllen. Daraus entstand ein neuartiges kryptografisches Zugriffskontrollsystem, das vor allem bei der Verarbeitungsgeschwindigkeit beim Schlüsselmanagement und der Sicherung (Backup) der geheimen kryptografischen Schlüssel besticht. Des Weiteren erfordert CASSP keine Neuverschlüsselung bei der Änderung von Zugriffsberechtigungen, die in der Praxis häufig auftritt, und entlastet damit extrem die Endgeräte und die Übertragungswege. Eine gesamte Neuverschlüsselung großer Datenmengen, wie sie z.B. bei Bildern sehr rasch auftreten kann, stellt für Smartphones und die Datenübertragung eine große Herausforderung dar, die mit CASSP bei Änderungen von Berechtigungen grundsätzlich entfällt.

Das Grundkonzept von CASSP basiert darauf, dass nicht verhindert werden kann, dass die zu schützenden Daten in der Cloud zerstört oder gelöscht werden (dies ist Aufgabe des Informati-

onssicherheitskonzepts des Cloud-Anbieters und des Zugriffskontrollsystems des Cloudspeicher-Systems). CASSP stellt aber sicher, dass die Daten von keinem unberechtigten Dritten gelesen oder gezielt verändert werden können.

Nachfolgend werden der Eigentümer bzw. Administrator der zu schützenden Daten als „*Eigentümer*“ bezeichnet und alle anderen Personen, die ebenfalls Zugriff (Lese-/Schreibberechtigung) zu den Daten erhalten sollen, als „*Partner*“ bezeichnet.

CASSP enthält einen Administrations-Teil, der unter der Hoheit des *Eigentümer*s liegt und in dem spezifische Daten dem *Eigentümer* und den jeweiligen *Partnern* zugeordnet werden. Da der Administrations-Teil von CASSP keine innovativen und kryptografisch interessanten Anteile enthält, wird er im Artikel nicht weiter behandelt. Dies gilt ebenfalls für die Datenverschlüsselung und Entschlüsselung. Es wird dabei der AES-Algorithmus im CTR Modus verwendet - in der Testversion mit einer Schlüssellänge von 128 Bit.

Die innovativen und kryptografisch interessanten Teile liegen vor allem im Schlüsselmanagement und der Sicherung (Backup) der beiden geheimen Zahlen p und q.

Nachfolgend wird unter dem Begriff „Administrator des Eigentümers“ der Administrations-Teil der CASSP-Software und unter „*Eigentümer*“ die gesamte CASSP-Software im Smartphone/Tablet des *Eigentümer*s bezeichnet. Der Begriff „*Partner*“ bezeichnet die gesamte CASSP-Software im Smartphone/Tablet des *Partners*.

4.1 Schlüsselmanagement

Das Schlüsselmanagement geht von folgenden Annahmen / Festlegungen aus:

- Der *Eigentümer* vergibt und verwaltet für alle *Partner* alle Zugriffsberechtigungen zum Cloudspeicher. Die Zugriffsberechtigung innerhalb von CASSP bezieht sich immer auf Lesen und Schreiben, weil CASSP auf einer symmetrischen Datenverschlüsselung aller Daten basiert (d. h. Lese- und Schreibberechtigung kann nicht unterschieden werden). Nur das Informationssicherheitskonzept des Cloud-Anbieters und Zugriffskontrollsystem des Cloudspeicher-Systems berücksichtigen weitere Attribute wie z.B. Lesen, Schreiben, Ändern, Löschen. Dies reicht, weil CASSP als kryptografisches Zugriffskontrollsystem sich nur vor einem unberechtigten Zugriff beim Cloudspeicher und auf dem Übertragungsweg schützt. Für die weiteren Zugriffsattribute reicht das Zugriffskontrollsystem beim Cloudspeicher.
- Die Ver- und Entschlüsselung aller Daten erfolgt mit dem symmetrischen Verschlüsselungsverfahren AES im CTR-Modus und dem kryptografischen Schlüssel z_i.
- Es existieren im gesamten System nur zwei geheime Parameter (p und q), die Primzahlen darstellen. Nur der *Eigentümer* kennt die geheimen Werte von p und q, die für die Ermittlung einer neuen Schlüsselversion ($z_{i+1,n}$) erforderlich sind. Für die Ver- und Entschlüsselung der Daten ist dann aber nur z_{i+1} erforderlich.
- Wenn ein neuer *Partner* eine Zugriffsberechtigung erhält, übermittelt ihm der *Eigentümer* in verschlüsselter Form den aktuellen Schlüssel ($z_{i+1,n}$).
- Mit diesem kryptografischen Schlüssel kann der *Partner* aber nur alle neuen Daten und Datenänderungen seit der letzten Schlüsseländerung lesen und, falls er dazu berechtigt ist, neue Daten bzw. Änderungen von vorhandenen Daten schreiben. Für alle älteren Daten benötigt er die früheren Schlüsselversionen (($z_{i-1,n}$), ($z_{i-2,n}$), …, ($z_{1,n}$)). Durch diesen

Mechanismus wird bei Änderungen von Zugriffsberechtigungen keine Neuverschlüsselung aller Daten erforderlich.

- Jeder *Partner* kann aus dem aktuellen Schlüssel (z_i,n) selbst und schnell alle älteren Versionen (z_{i-1},n), (z_{i-2},n), …, (z_1,n) ermitteln.

Nach einer ausführlichen wissenschaftlichen Recherche und Analyse zeigte sich, dass diese Anforderungen an das Schlüsselmanagement besonders gut durch modulares Quadrieren (*mod n* mit $n=p\cdot q$) und Quadratwurzelziehen gelöst werden können. Das Quadrieren benötigt kein Geheimnis (nur n muss bekannt sein) und geht sehr schnell und für das Wurzelziehen ist ein Geheimnis erforderlich (p und q). Für die Quadratwurzel existieren mehrere Methoden, wie z.B. das Verfahren von Adlemann, Manders und Miller, das wiederum auf Cipolla (1903) basiert und das anwendbar ist für alle Primzahlen p und q, oder ein Verfahren, das beim Rabin-Kryptosystem angewendet wird und gilt nur für: $p \equiv q \equiv 3 \pmod 4$.

Beide Verfahren des modularen Quadratwurzelziehens sind sehr zeiteffizient umzusetzen und wurden umfangreich getestet. Für CASSP wurde entschieden, dafür die Ideen des Rabin-Kryptosystems [Rabi79, Bart12] zu übernehmen. Dabei konnte der Hauptnachteil des Rabin-Krypto-Systems, die vier Ergebnisse beim Wurzelziehen (bei der Ver-/Entschlüsselung möchte man immer nur ein Ergebnis haben), vermieden und sogar zu einem Vorteil umgewandelt werden, weil bei CASSP die Quadratwurzel ein Quadratischer Rest *mod n* sein muss und dieser Fall tritt bei vier Ergebnissen immer ein - bei nur einem einzigen Ergebnis hätte es hier Probleme geben können.

In CASSP bedeutet dies, dass ausgehend von einer Zahl z, die in der ersten Schlüsselgeneration eine Zufallszahl darstellt (d.h. z_1 = Zufallszahl), und zwei geheimen Primzahlen p und q mit den Bedingungen "$p \equiv q \equiv 3 \pmod 4$", die nur der *Eigentümer* kennt, die Berechnung eines neuen kryptografischen Schlüssels (z_{i+1},n) für $i>0$ durch eine modulare Quadratwurzel von z_i *(modulo n)* mit $n = p\cdot q$ erfolgt. Die Berechnung von z_{i+1}, d.h. der modularen Quadratwurzel von z_i *(modulo n)*, erfolgt durch die Formeln

$a_1 = \sqrt{z_i} \equiv z_i^{(p+1)/4} \pmod p$ und $b_1 = \sqrt{z_i} \equiv z_i^{(q+1)/4} \pmod q$

$a_2 = \sqrt{z_i} \equiv p - a_1 \pmod p$ und $b_2 = \sqrt{z_i} \equiv q - b_1 \pmod q$

und dann mit Hilfe des Chinesischen Restsatzes $z_{i+1} = \sqrt{z_i} \equiv b \cdot s \cdot p + a \cdot t \cdot q \pmod n$

Die Werte von s und t werden mit Hilfe des Erweiterten Euklidischen Algorithmus aus p und q ermittelt. Aus dem Chinesischen Restsatz ergeben sich durch die vier Möglichkeiten von a und b ((a_1,b_1), (a_1,b_2), (a_2,b_1) und (a_2,b_2)) vier verschiedene Ergebnisse der modularen Quadratwurzel. Da der generierte neue Schlüssel z_{i+1} ein Quadratischer Rest *(modulo n)* sein muss, wird das erste (von den vier) Ergebnissen in aufsteigender Reihenfolge als z_{i+1} ausgewählt, das diese Bedingung erfüllt. Zur Feststellung des quadratischen Rests wird das Euler'sche Kriterium ($z_{i+1}^{(p-1)/2} \pmod n = 1$) angewendet. Da n mehr als 1.500 Bit aufweist, wird in CASSP die Multiplikation mit diesen großen Bitlängen aus Gründen der Zeiteffizienz mit Hilfe der Buminov Methode [BuSc05, ScBu04], die auf Montgomery basiert, umgesetzt.

Der neue kryptografische Schlüssel (z_{i+1},n) wird vom Administrations-Teil des mobilen Geräts des *Eigentümers* an die mobilen Geräte aller *Partner* in verschlüsselter Form (unter Verwendung des RSA-Algorithmus) gesendet. Für diese Verschlüsselung senden vorher alle *Partner* ihr dafür vorgesehenes Zertifikat, das auch die Adresse des *Partners* enthält und vom *Eigentümer* signiert wurde, an den *Eigentümer*. Damit sollen Man-in-the-middle-Angriffe verhindert

werden. Durch modulares Quadrieren können dann alle *Partner* alle früheren Schlüsselgenerationen jederzeit selbst rasch und zeiteffizient berechnen. Dazu benötigen sie nur den Wert von n (der nicht geheim ist) und nicht die geheimen Werte von p und q.

Die Erzeugung einer neuen Schlüsselgeneration (z_{i+1},n) aus (z_i,n) ist immer dann erforderlich, wenn ein *Partner* ausscheidet (seine Zugriffsrechte verliert). Alle anderen *Partner* sollen natürlich weiterhin Zugriff auf die Daten haben. Da der ausgeschiedene *Partner* auf die neuen Daten bzw. alle Änderungen von vorhandenen Daten keinen Zugriff mehr bekommen soll, erhalten alle noch berechtigten *Partner* einen neuen geheimen Schlüssel. Die alten (bisherigen) Daten bleiben unverändert, d.h. sie werden aus Zeiteffizienzgründen – insbesondere zur Entlastung des Smartphones und Übertragungsweges - nicht neu verschlüsselt. Es besteht auch kein Grund dafür, weil der ausgeschiedene *Partner* auch vorher Zugriff zu diesen Daten hatte. Nur der *Eigentümer* kann aus z_i ein neues z_{i+1} ermitteln – nur er kennt p und q - und alle noch berechtigten *Partner* können aus dem neuesten z_i alle älteren Generationen einfach und schnell ohne Kenntnis von p und q durch modulares Quadrieren ermitteln, denn es gilt $z_{i-1} = z_i^2 \ (mod\ n)$ für $i > 2$.

Dieser Vorgang ist vor allem für einen neuen *Partner* wichtig, weil er vom *Eigentümer* nur die neueste Schlüsselgeneration erhält und mit diesem Schlüssel nur alle neuen Daten und Datenänderungen seit der letzten Schlüsseländerung lesen kann und, falls er dazu berechtigt ist, neue Daten schreiben bzw. Datenänderungen durchführen kann. Für alle älteren Daten benötigt er alle früheren Schlüsselversionen (z_{i-1},n), (z_{i-2},n), …, (z_1,n), die er nur durch modulares Quadrieren und ohne Kenntnis der geheimen Daten p und q ermitteln kann.

4.2 Schlüsselsicherung

Ein wichtiger Teil von CASSP ist auch die Sicherung (Backup) der geheimen Daten p und q, die für jede Neuberechnung des Schlüssels (z_{i+1},n) aus (z_i,n) erforderlich sind, und der Zufallszahl z_1, aus der sich mit Hilfe von p und q alle z_i (für $i > 1$) und damit Schlüsselgenerationen ermitteln lassen (siehe oben). Die Sicherung auf einem externen Datenträger kann Probleme darstellen, weil z.B. externe Datenträger nicht verwendbar sind (wie z.B. USB-Sticks) oder vom Benutzer für eine Datensicherung nicht verwendet werden (wie z.B. Mikro-SD-Karten). Eine verschlüsselte Sicherung in der Cloud, wo auch die eigentlichen Daten liegen, verlagert das Problem auf einen weiteren kryptografischen Schlüssel und löst damit nicht das Problem. Die in CASSP erarbeitete Lösung baut auf einem (k, n)-Schwellwertschema [RoDe82, Sham79] auf.

p, q und z_1 werden nachfolgend Basisschlüssel genannt, weil aus diesen drei Werten alle kryptografischen Schlüssel ermittelt werden können. Das hat Nachteile:

- Wenn der Basisschlüssel zufällig oder böswillig bekannt wird, dann ist das ganze System verwundbar.
- Wenn der Basisschlüssel verlorengeht (Verlust des Gerätes, Hardwaredefekt etc.) oder gelöscht wird, dann können sämtliche Informationen unzugänglich werden.

Der Verlust des Basisschlüssels kann gelöst werden, indem:

a) Kopien an "vertrauenswürdige" Nutzer gegeben werden oder

b) die aktuelle Schlüsselgeneration von einem *Partner* (sofern welche vorhanden sind) an den *Eigentümer* zurückübertragen wird.

Im Fall a) wird das System jedoch anfällig gegen "böswillige Freunde" oder ggf. auch einen neugierigen Betreiber. Im Fall b) können dann keine neuen Schlüsselgenerationen mehr ermittelt werden, d.h. es muss der *Eigentümer* einen neuen Basisschlüssel (neues p, q und z_1) ermitteln und mit diesem alle Daten im Cloudspeicher neu verschlüsseln.

Die Lösung in CASSP besteht darin, den Basisschlüssel *d* (bestehend aus *p*, *q* und z_1) in *n* Teile $d_1, d_2, \ldots, d_n$ so zu zerlegen, dass

1. bei Kenntnis beliebiger *k* Teile d_i, d einfach berechnet werden kann und,
2. bei Kenntnis von *k-1* oder weniger Teile d_i, *d* mangels Informationen nicht bestimmt werden kann.

Die *n* Teile werden an *n Speicherorte* übertragen. *Speicherorte* können dabei verschiedene Dateien in gleichen bzw. verschiedenen Cloudspeichern oder externen Speichermedien sein. Weil *k* Teile für die Wiederherstellung des Basisschlüssels nötig sind, gefährdet die Bekanntheit eines Teils oder von bis zu *k-1* Teilen nicht das Geheimnis des Basisschlüssels. Der Basisschlüssel kann auch dann wieder hergestellt (ermittelt) werden, wenn ein Teil verlorengeht oder zerstört wird (solange es wenigstens *k* intakte Teile gibt). Blakley [GR79] veröffentlichte das erste Schwellwertschema, welches auf projektiver Geometrie beruhte. Ein von Shamir vorgeschlagenes Schema beruht auf Polynominterpolation und dem Fundamentalsatz der Algebra [Sham79]. Man benötigt nämlich mindestens *k* Wertepaare, um ein Polynom vom Grad *k-1* eindeutig zu bestimmen. Blakley zeigte, dass das Schema modulo einem irreduziblen Polynom der Form $x^n + x^1 + 1$ (nicht für jedes *n* irreduzibel!) effizient implementiert werden kann [GR79, GR80]. Eine sehr effiziente Implementierung dieses Schemas findet man in [Poet].

Um eine Verfälschung der Teile $d_1, d_2, \ldots, d_n$ sicher zu erkennen, werden vom *Eigentümer* alle Teile digital signiert. Der Basisschlüssel (p, q, z_1) stellt dabei den verwendeten geheimen Signaturschlüssel dar, denn geprüft wird die digitale Signatur mit dem öffentlichen Signaturschlüssel. Dadurch wird ein weiterer geheimer Schlüssel eingespart, der wieder gesichert werden müsste.

Um unterschiedlich vertrauenswürdige *Speicherorte* entsprechend zu gewichten, kann im System *k* und *n* vergrößert werden, so dass vertrauenswürdige *Speicherorte* mehrere Teile, weniger vertrauenswürdige hingegen nur einen Teil erhalten.

In CASSP wird mit den eigentlichen Daten zumindest immer auch ein Teil (der insgesamt *n* Teile $d_1, d_2, \ldots, d_n$) als Speicherort verwendet und an dieser Stelle auch der öffentliche Signaturschlüssel und alle digitalen Signaturen über die einzelnen Teile $d_1, d_2, \ldots, d_n$ zur späteren Prüfung dieser digitalen Signaturen abgespeichert.

5 Diskussion und Ausblick

In diesem Beitrag wird gezeigt, dass bei kryptografischen Zugriffskontrollsystemen noch viel Bedarf nach Forschung und neuer / verbesserter Lösungen am Markt besteht. Den umfangreichen Forschungs- und Entwicklungstätigkeiten im Forschungsprojekt „Smartphone Security“, wo CASSP nur eines von drei zentralen Ergebnissen darstellt, werden am Institut für IT Sicherheitsforschung der FH St. Pölten in Zukunft sicher noch einige weitere folgen. Das innovative Potenzial ist in diesem Bereich noch groß und vielfältig.

Für CASSP wurde ein Prototyp für Tablets und Smartphones auf Android-Plattform entwickelt und ausführlich getestet. Die Testergebnisse sind in Bezug auf die Verarbeitungsgeschwindigkeit sehr gut, was darauf hinweist, dass das ausgearbeitete Systemkonzept, die ausgewählten

Algorithmen und die Implementierung reibungslos zusammenwirken. Da CASSP im Rahmen eines Forschungsprojektes entstand, wurde bei der Entwicklung relativ wenig Rücksicht genommen auf Benutzerfreundlichkeit und rechtliche Aspekte der ausgewählten Algorithmen und Verfahren. Diese Punkte sind zu berücksichtigen, wenn man den Schritt zu einem fertigen Produkt wählt, was in Zukunft geplant ist.

Bei CASSP erfolgt nach dem Entzug einer Leseberechtigung keine Neuverschlüsselung (was die Endgeräte in hohem Maße entlastet). Rechtlich kann das aber problematisch sein. Aus urheberrechtlicher Sicht gelten die manchmal geschützten Daten, wenn ein *Partner* die Gruppe verlässt und deshalb eine neue Schlüsselgeneration erzeugt wird, als der Öffentlichkeit zur Verfügung gestellt (§ 18a UrhG AT) bzw. als öffentlich zugänglich gemacht (§ 19a UrhG DE), was zu den Ausschließlichkeitsrechten des Rechtsinhabers zählt. Abhilfe schafft hier nur die Löschung bzw. Neuverschlüsselung der betroffenen Daten, um die künftige Nutzung durch Externe auszuschließen. Aus deutscher datenschutzrechtlicher Sicht werden die Daten, soweit sie personenbezogen sind, auch an Außenstehende übermittelt (nach § 3 Abs. 4 Nr. 3 lit. b BDSG, wenn „Dritte zur Einsicht oder zum Abruf bereitgehaltene Daten einsehen oder abrufen“), was sich, soweit es nicht ohnehin ausschließlich für persönliche oder familiäre Tätigkeiten erfolgt, auf gleiche Weise beenden lässt.

Literatur

[Bart12] G. Barthe et al. Verified Security of Redundancy-free Encryption from Rabin and RSA. In Proc. of the 2012 ACM Conference on Computer and Communications Security, CCS '12, pp. 724-735, NY, USA, 2012

[BuSc05] V. Bunimov and M. Schimmler. Completely redundant modular exponentiation by operand changing. In Proc. of the 2005 International Conference on Computer Design, CDES 2005, Las Vegas, USA, 2005, pp. 224-232

[Mats08] T. Matsunaka et al. Secure Data Sharing in Mobile Environments. In Proc. of the 9th International Conference on Mobile Data Management, MDM '08, pp. 57-64, Washington, DC, USA, 2008

[HaJe03] A. Harrington and C. Jensen. Cryptographic Access Control in a Distributed File System. In Proc. of the 8th ACM Symposium on Access Control Models and Technologies, SACMAT '03, pp. 158 - 165, NY, USA, 2003.

[TC14] TrueCrypt. Containerbasiertes Verschlüsselungssystem. http://truecrypt.sourceforge.net/, 2014.

[Jaeb12] S. Jaebok, et al. "DFCloud: A TPMbased secure data access control method of cloud storage in mobile devices." Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on. IEEE, 2012.

[Sasc12] F. Sascha et al. "Confidentiality as a ServiceUsable Security for the Cloud." Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on. IEEE, 2012.

[Geer13] S. Geerlings. "Measurements for the Paranoid: The Effect of Encrypting Files in Cloud Storage", University of Twente, The Netherlands, 2013.

[Half14] G. Halfacree. “TrueCrypt downed by alleged insecurities”. http://www.bittech.net/news/bits/2014/05/29/truecryptdown/, 2014.

[Enci14] EncipherIt. Online Verschlüsselungssystem. https://encipher.it/, 2014.

[Boxc14] Boxcryptor. Dateibasiertes Verschlüsselungssystem für CloudSpeicher. https://www.boxcryptor.com/, 2014.

[PaSo13] C Patsakis and A. Solanas. "Trading Privacy in the Cloud: A Fairer Way to Share Private Information." eBusiness Engineering (ICEBE), 2013 IEEE 10th International Conference on. IEEE, 2013.

[Stutt] http://www.stuttgarter-zeitung.de/inhalt.icloud-panne-apple-keine-sicherheitsluecke.f06f97e4-e86a-4a4e-a671-2a4ae082f69b.html

[Rabi79] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, Cambridge, MA, USA, 1979.

[ScBu04] M. Schimmler and V. Bunimov. Fast modular multiplication by operand changing. In International Conference on Information Technology: Coding and Computing (ITCC'04), Volume 2, 2004, Las Vegas, USA, pp. 518-524, 2004

[RoDe82] D. E. Robling Denning. Threshold Schemes. In Cryptography and Data Security, page 179. Addison-Wesley Longman Publishing Co., Inc., Boston, USA, 1982

[Sham79] A. Shamir. How to Share a Secret. Communication ACM, 22(11): pp 612-613, 1979

[GR79] GR Blakley: Safeguarding cryptographic keys, Proc. AFIPS 1979 NCC, vol 48, June 1979, pp. 313-317

[GR80] GR Blakley. One-Time Pads are Key Safeguarding Schemes, Not Cryptosystems, Proceedings of 1980 Symposium on Security and Privacy IEEE, April 1980, pp. 108-113

[Poet] B. Poettering. Implementation of Shamir's Secret Sharing Scheme (GPL) http://point-at-infinity.org/ssss/ssss-0.5.tar.gz

[Chu13] C. K.Chu et al. “Security Concerns in Popular Cloud Storage Services.” IEEE pervasive computing 12.4 (2013): 5057.