Sichere und effiziente Zugriffskontrolle mit PAMINA

Zoltán Nochta und Sebastian Abeck

Universität Karlsruhe, Institut für Telematik
nochta@cm-tm.uka.de abeck@cm-tm.uka.de

Abstract: Der Beitrag stellt PAMINA (Privilege Administration and Management INfrAstructure) vor, ein System für die zertifikatsbasierte Autorisierung und Zugriffskontrolle. PAMINA verwaltet die Zertifikate mit Improved Certification Verification Trees (I-CVT). I-CVTs beruhen auf den in [GGM00] vorgeschlagenen CVTs, sie bieten aber bessere Performanz und die Möglichkeit, Verifizierern die Vollständigkeit der von einer nicht vertrauenswürdigen Datenbank erhaltenen Rechte zu beweisen. Das System ermöglicht Benutzern, ihre Privilegien an Andere zu delegieren, wobei Verifizierern die ansonsten aufwändige Prüfung von Delegierungsketten erspart bleibt.

1 Einführung

Die zunehmende Nutzung sicherheitskritischer elektronischer Dienste erfordert die Schaffung von Systemen, die Ressourcen gegen unautorisierte Zugriffe schützen. Ein Zugriffskontrollsystem gestattet einem bereits authentifizierten Subjekt (Person, Objekt, usw.) den Zugriff auf Ressourcen (Objekte, Dateien, usw.), nachdem es festgestellt hat, dass das Subjekt die benötigten Rechte besitzt. Das Ziel ist, nur autorisierte Zugriffe zuzulassen und alle unautorisierten Zugriffe zu verhindern.
In verteilten Systemen erfolgt die Zugriffskontrolle seitens des Zielsystems (Server), indem ein Verifizierungsprogramm die ihm aktuell zur Verfügung stehenden Autorisierungsdaten (Rechte, Richtlinien, usw.) entsprechend auswertet. Das Managementsystem muss u.a. dafür sorgen, dass ein Verifizierer die Originalität, Gültigkeit und Vollständigkeit der Daten *nach* deren Erstellung zweifelsfrei prüfen kann. Diese Anforderungen treten unabhängig vom jeweiligen Administrationsmodell (ACLs, Rollen, usw.) auf. Digital signierte Autorisierungszertifikate bieten zwar einen guten Schutz vor Manipulation, sie machen aber die Verifizierung aufwändiger, und es entsteht ein nicht unerheblicher Zusatzaufwand auch seitens der ausstellenden Autorität.
Wichtige Ziele von PAMINA sind, diese Zusatzkosten zu reduzieren und die Zugriffsentscheidungen einzig allein vom Willen der vertrauten Stellen (*Privilege Management Authorities, PMAs*) abhängig zu machen. Die PMAs verwalten und signieren ihre Daten (I-CVTs) lokal und senden die Änderungen regelmäßig an eine Datenbank, die diese für Verifizierer, PMAs und End-Benutzer zur Verfügung stellt (s. Abb.1). Die Datenbank ist nicht in der Lage, einzelne Zertifikate zu verändern, und sie muss auf eine Anfrage alle zutreffenden Zertifikate ausliefern.

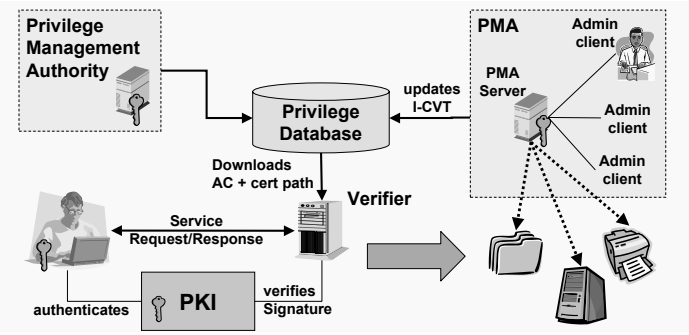

Abbildung 1: Systemkomponenten und ihre Interaktion

Die prototypische Implementierung realisiert das Server-Pull Modell, welches performanter ist als der Client-Push Ansatz (s. dazu [NEA02]), und integriert die Dienste einer kommerziellen PKI.
In den weiteren Abschnitten des Beitrags stellen wir zunächst I-CVTs vor und diskutieren Aspekte der Rechtedelegierung.

2 Das zugrunde liegende Zertifikatsmanagementschema

Zertifikate haben in PAMINA eine kurze Gültigkeitsperiode (z.B. eine Stunde), sie werden alle regelmäßig von den PMAs aktualisiert (neu signiert). Die Tatsache, dass in der Datenbank nur gültige oder nur abgelaufene Zertifikate einer PMA vorhanden sind, macht die Verifizierung wesentlich einfacher und reduziert die Verwaltungskosten seitens der PMA, die kein Sperrsystem verwenden muss.
Eine Datenstruktur mit diesen Eigenschaften, der sog. *Certification Verification Tree (CVT)*, wurde in [GGM00] vorgeschlagen. Ein CVT ist ein Merkle-Hashbaum, in dessen Blättern unsignierte Zertifikate, sog. *Cert-Statements (CS)*, und jeweils ein Hashwert gespeichert werden. Nur der Hashwert des Wurzelknotens wird mit zusätzlichen Informationen, wie z.B. Gültigkeitsperiode, signiert. Ein zu sperrendes Zertifikat kann aus dem Baum einfach gelöscht werden.
Die Basis eines Improved CVT (I-CVT) bildet ein B^+-Baum, der u.a. eine effiziente sequentielle Suche nach Zertifikaten ermöglicht. Ein wichtiges Ziel bei der Konstruktion der I-CVTs war, einen Vollständigkeitsbeweis der ausgelieferten Zertifikate effizient liefern zu können. Ein I-CVT wird wie folgt konstruiert: Dem Hashbaum bzw. den Zertifizierungspfaden (*Certification Path, CP*) werden Informationen über die Struktur des B^+-Baums hinzugefügt [BLL00]. In jedem Knoten v wird ein Hashwert $H[v]$ gespeichert, welcher der Hashwert über die Suchschlüssel $k_i[v]$ in diesem Knoten und die Hashwerte seiner Kinder ist (s. Abb. 2). Diese Technik ermöglicht die Konstruktion von CPs, durch die Verifizierer auch die Vollständigkeit der erhaltenen Daten prüfen können. Abb. 2 zeigt einen I-CVT, in dem Benutzer mehrere Zertifikate besitzen, die durch ein $(Benutzername, Seriennummer)$ Paar zu unterscheiden sind.

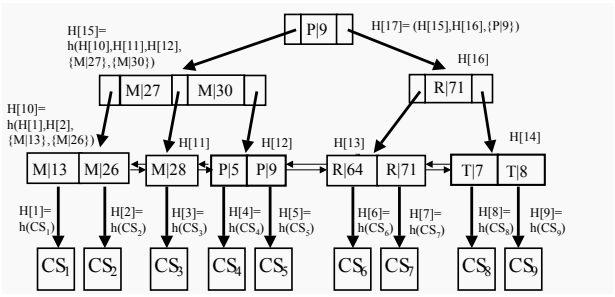

Abbildung 2: Beispiel I-CVT mit mehreren Zertifikaten pro Benutzer

Die Zertifikate eines Benutzers bilden einen abgeschlossenen Bereich innerhalb des Baums. Man kann zeigen, dass es wegen letzterer Eigenschaft und der doppelten Verkettung der Blätter ausreicht, einem anfragenden Verifizierer, der alle Zertifikate eines Benutzers haben möchte, einen einzigen CP zu senden, der die Vollständigkeit der erhaltenen Daten bestätigt. Beachte, dass der Verifizierer in der in [BLL00] behandelten Konstruktion CP(CS_4) und CP(CS_5) prüfen müsste, um die Vollständigkeit der Rechte von P festzustellen. In unserem Fall sendet die Datenbank neben den unsignierten Zertifikaten CP(P):

$$CP(P) = (([P9], H[15], H[16]), ([M27], [M30], H[10], H[11], H[12]),$$
$$([M28], [P5], [P9], [R64], H[4], H[5]), ([R71], H[13], H[14]),$$
$$([P9], [R64], [R71], [T7], H[6], H[7]), \text{Gültigkeit}, \text{Signatur}(H[17], \text{Gültigkeit}))$$

Dieser CP bestätigt auch die Nicht-Existenz des Benutzers Q. Durch den Einsatz der I-CVTs können PMAs und Verifizierer sichergehen, dass die Datenbank stets korrekte und vollständige Daten ausliefert.

2.1 Bedeutung der Vollständigkeitsbeweise in der Zugriffskontrolle

Die eingeführten Vollständigkeitsbeweise, die die Speicherinstanz liefern kann und muss, reduzieren das Vertrauen in die Datenbank, weil sie die unbemerkte Manipulation der Menge der ausgelieferten Zertifikate verhindern. Sie schützen die Anwendung gegen Angriffe, die zu der Verweigerung autorisierter oder der Zulassung unautorisierter Zugriffe führen könnten. In einem Autorisierungssystem, welches positive Rechte (*permissions*) verwendet und die Vollständigkeit nicht garantiert, kann auf Anfrage eines Verifizierers z.B. ein von A bestochener Datenbankbetreiber alle Rechte von A an den Verifizierer senden, und die ansonsten existierenden Rechte von B zurückhalten, und ihn somit vom berechtigten Zugriff fernhalten.
Manche Systeme erlauben auch die Verwendung von negativen Rechten (*denials*). Informell ausgedrückt besagt ein negatives Recht, was ein bestimmtes Subjekt in dem System nicht tun darf. Bei fehlender Prüfung der Vollständigkeit kann nun die unehrliche oder manipulierte Datenbank die Existenz einiger (aller) negativer Zugriffsrechte von A abstreiten. In diesem Fall könnte also A unerlaubt zugreifen.

3 Effiziente Rechtedelegierung

Die Rechtedelegierung ist das Übertragen einer Teilmenge der Rechte des Delegierungsgebers (DG) auf den Delegierungsempfänger (DE). Um die weitergegebenen Rechte und deren Gültigkeitsperiode vor Manipulation zu schützen, kann der DG Zertifikate ausstellen. Durch mehrstufige Delegierung können die Zertifikate ein Netzwerk bilden, welches im einfachsten Fall eine Kette ist. Ein Verifizierer muss bei einem Zugriffsversuch feststellen, ob das benötigte und evtl. mehrmals weiterdelegierte Recht seinen Ursprung bei der vertrauten Instanz (in unserem Fall einer PMA) hat. Hierzu muss er alle Zertifikate im Delegierungsnetzwerk prüfen. In einem Server-Pull (Client-Push) System muss die Rechtedatenbank (der Zugreifende) die relevanten Zertifikate suchen und an den Verifizierer senden, was sehr zeitaufwändig sein kann. Das Einfügen der Delegierungsgeschichte ([NEA02]) in die Zertifikate beschleunigt zwar die Suche, aber wenn z.B. ein Zertifikat gesperrt (gelöscht) wird, müssen alle hiervon betroffenen Zertifikate geändert werden. In [Au99] wurde die elegante Methode der Zertifikatsreduktion vorgestellt: Der Ressourceninhaber ersetzt die Delegierungsketten jeweils durch ein einziges Zertifikat. Unklar bleibt dabei u.a. wie in einem verteilten System nach den zu reduzierenden Ketten gesucht wird und wie die DGs über die durchgeführte Reduktion benachrichtigt werden.
Um solche verwaltungstechnischen Probleme zu vermeiden, bietet der PMA-Server eine Schnittstelle an, über die autorisierte PMAs und Benutzer einen signierten Delegierungsantrag stellen können. Nach dem Erhalt eines Antrags prüft der PMA-Server, ob der potenzielle DG die ausreichenden Rechte besitzt und stellt direkt ein Zertifikat für den vorgesehenen DE aus. Der PMA-Server führt ein Buch über die Delegierungsgeschichten und, falls ein DG die Rechte widerrufen möchte, löscht er die hiervon betroffenen Zertifikate aus dem eigenen I-CVT. Dieser Ansatz reduziert den Aufwand bei der Verifizierung erheblich, ohne die Sicherheit des Systems durch unkontrollierbare Delegierung zu gefährden.

Literaturverzeichnis

[Au99] T. Aura. Distributed Access-Rights Management with Delegation Certificates. Security Issues for Distributed and Mobile Objects, LNCS 1603, S. 211-235, Springer, 1999.

[BLL00] A. Buldas, P. Laud, H. Lipmaa. Accountable Certificate Management using Undeniable Attestations. Proc. of the 7th ACM Conference on Computer and Communication Security, S. 9-17, Athens, Greece. November 2000.

[GGM00] I. Gassko, P. Gemmell, P. MacKenzie. Efficient and Fresh Certification. Public Key Cryptography ’2000, LNCS 1751, S. 342-353, Springer, 2000.

[NEA02] Z. Nochta, P. Ebinger and S. Abeck. PAMINA: a Certificate Based Privilege Management System. Network & Distributed System Security Symposium, S. 167-179, San Diego, USA. Februar, 2002.