Armin B. Cremers, Rainer Manthey,
Peter Martini, Volker Steinhage (Hrsg.)

Sicherheit in komplexen, vernetzten Umgebungen

Workshop im Rahmen der Jahrestagung 2005 der Gesellschaft für Informatik
„Informatik LIVE!“

19.-22. September 2005
in Bonn, Deutschland

Gesellschaft für Informatik 2005

Lecture Notes in Informatics (LNI) – Proceedings
Series of the Gesellschaft für Informatik (GI)

Volume P-68

ISBN 3-88579-397-0
ISSN 1617-5468

Volume Editors
Prof. Dr. Armin B. Cremers
Universität Bonn, Institut für Informatik III
Römerstraße 164, D-53117 Bonn
Email: abc@cs.uni-bonn.de
Prof. Dr. Rainer Manthey
Universität Bonn, Institut für Informatik III
Römerstraße 164, D-53117 Bonn
Email: manthey@cs.uni-bonn.de
Prof. Dr. Peter Martini
Universität Bonn, Institut für Informatik IV
Römerstraße 164, D-53117 Bonn
Email: martini@cs.uni-bonn.de
Privatdozent Dr. Volker Steinhage
Universität Bonn, Institut für Informatik III
Römerstraße 164, D-53117 Bonn
Email: steinhage@cs.uni-bonn.de

printed by Köllen Druck+Verlag GmbH, Bonn

Public-Key-Infrastrukturen in einer Peer-to-Peer-Umgebung

Thomas Wölfl, Sven Wünschmann

Universität Regensburg
Institut für Wirtschaftsinformatik
Thomas.Woelfl@wiwi.uni-regensburg.de
Sven.Wuenschmann@stud.uni-regensburg.de

Abstract: Der Einsatz von Peer-to-Peer (P2P) Netzwerken verspricht im Vergleich zu Client-Server-Netzwerken eine bessere Skalierbarkeit, niedrigere Betriebskosten, die Nutzung bislang unausgelasteter Ressourcen und höhere Fehlertoleranz. Eine wichtige Eigenschaft von P2P-Netzwerken ist der Verzicht auf zentrale, vertrauenswürdige Instanzen. Die Realisierung einer Public-Key-Infrastruktur (PKI) basierend auf modernen P2P-Techniken erfordert deswegen besondere Sicherheitsmaßnahmen. Es ist das Ziel dieser Arbeit, die Vorteile einer P2P basierten PKI vorzustellen und geeignete Sicherheitsmechanismen zu präsentieren.

1 Einleitung

Um Informationssicherheit in einer Internetumgebung zu erreichen, stehen die modernen Methoden der Public-Key-Kryptographie zur Verfügung. Die Anwendbarkeit dieser Methoden steht und fällt jedoch mit der zuverlässigen Bindung der Identität eines Subjekts an dessen kryptographischen öffentlichen Schlüssel. Es ist das Ziel einer Public-Key-Infrastruktur (PKI), diese Bindung sicherzustellen.

Zertifizierungsstellen stellen elektronische Public-Key-Zertifikate aus, um die zuverlässige Bindung des öffentlichen Schlüssels an das zugehörige Subjekt zu bescheinigen. Ein Public-Key-Zertifikat ist ein von der Zertifizierungsstelle digital signierter String, der den Namen des Benutzers, den zugeordneten öffentlichen Schlüssel, Beginn und Ende der Gültigkeit des Zertifikats, den Namen des Ausstellers und weitere technische Informationen enthält.

Gleichzeitig zeichnet sich eine Entwicklung von Client-Server-Architekturen hin zur Peer-to-Peer (P2P) Architektur ab. Ein P2P-Netzwerk ist ein Verbund von Gleichberechtigten (Nodes), die sich wechselseitig Ressourcen unter Verzicht auf zentrale Koordinationsinstanzen zur Verfügung stellen (vgl. [SF02]). Der Einsatz von P2P-Netzwerken verspricht eine bessere Skalierbarkeit, niedrigere Betriebskosten, die Nutzung bislang unausgelasteter Ressourcen und höhere Fehlertoleranz.

Die Realisierung einer PKI mit Hilfe eines P2P-Netzwerks bietet eine Reihe von Vorteilen. Hierbei sind Sicherheitsmaßnahmen für die zuverlässige Übertragung von Public-Key-Zertifikaten in einem P2P-Netzwerk einzuführen. Die beteiligten Nodes des P2P-Netzes

sind keine vertrauenswürdigen Instanzen. Sie könnten Zertifikate zurückhalten und somit die Verfügbarkeit des PKI-Dienstes gefährden. In den folgenden Abschnitten werden zunächst die angesprochenen Vorteile einer P2P-PKI vorgestellt. Anschließend werden Sicherheitsmechanismen präsentiert, welche die Verfügbarkeit von Zertifikaten absichern.

Es liegen zwei Arbeiten aus dem Bereich P2P-PKI vor. Datta et al. [DHA03] präsentieren eine P2P basierte PKI, die das Ziel verfolgt, die aktuelle IP-Adresse eines Systems an einen öffentlichen Schlüssel zu binden. Diese Arbeit verwendet das P2P-Protokoll P-Grid [Abe01]. Einen alternativen Ansatz verfolgt die Arbeit [Wöl05]. Es werden Algorithmen zur Realisierung einer PKI vorgestellt, mit dem Ziel, die Identität eines Benutzers an einen öffentlichen Schlüssel zu binden. Letztere Arbeit verwendet zur Übertragung von Zertifikaten das strukturierte P2P-Protokoll Chord [SMLN$^+$03].

2 Vorteile einer P2P basierten PKI

2.1 Unabhängigkeit von einer Betreiberorganisation

Existierende Public-Key-Infrastrukturen werden nur in einem geringen Ausmaß von Internetanwendern akzeptiert. Ein Grund für diese geringe Akzeptanz liegt in der hohen Abhängigkeit vorhandener PKIs von Betreiberorganisationen. Für eine weite Verbreitung und Akzeptanz in einer Internetumgebung müssen PKIs entwickelt werden, die sich am Paradigma der Selbstverwaltung im Internet orientieren. Um erfolgreich zu sein, sollte eine PKI unabhängig von einer Betreiberorganisation sein.

Tabelle 1: Unabhängigkeit von Public-Key-Infrastrukturen

Anwendungsgebiet	Unabhängigkeit der Zertifizierung	Unabhängigkeit vom Verzeichnis	Beispiel
B2B, B2C	- (Hierarchisch)	- (Zentral)	Verisign, Thwate
C2C	+ (Web of Trust)	- (Zentral)	PGP
C2C	+ (Web of Trust)	+ (Dezentral)	P2P-PKI

Um klassische PKI-Ansätze und P2P-PKI Ansätze voneinander abzugrenzen, werden zwei Dimensionen der Unabhängigkeit von Public-Key-Infrastrukturen betrachtet (vgl. Tabelle 1). Die erste Dimension ist *Unabhängigkeit der Zertifizierung*. In einer PKI basierend auf einem hierarchischen Trust Model (vgl. [AL03]) dürfen nur dedizierte Zertifizierungsstellen Zertifikate ausstellen. Ein anderer PKI-Typ verwendet ein Web-of-Trust Trust Model. Hierbei kann jeder Teilnehmer Zertifikate ausstellen. Gleichzeitig muss jeder Teilnehmer individuell entscheiden, von wem er Zertifikate akzeptiert (Vertrauen). Die Zertifizierung in einem Web-of-Trust ist unabhängig von einer Betreiberorganisation.

Die zweite Dimension ist *Unabhängigkeit vom Verzeichnisdienst*, welcher dazu verwendet wird, Zertifikate zu speichern und abzurufen. Der größte Teil existierender PKIs verwendet ein zentrales Verzeichnis für die Speicherung von Zertifikaten. Die Organisation, die

dieses Verzeichnis kontrolliert, hat auch die Kontrolle über die Verteilung der Zertifikate. Um Unabhängigkeit zu erreichen, sollten Zertifikate dezentral gespeichert werden. Dezentralisierung ist einer der zugrunde liegenden Faktoren für den Erfolg von bedeutenden Internetdiensten wie zum Beispiel E-Mail (es gibt keinen zentralen E-Mail Server).

2.2 Ressourcenausnutzung und Fehlertoleranz

Die verteilte Speicherung von Zertifikaten hat zur Folge, dass ungenutzte Ressourcen wie Bandbreite und Speicherplatz besser ausgenutzt werden. In einem P2P-Netzwerk gibt es keinen Single Point of Failure. Der Ausfall einer Node betrifft nur die Node selbst und nicht den gesamten PKI-Dienst.

2.3 Lastverteilung und Skalierbarkeit

Eine weit verteilte PKI muss eine enorme Anzahl von (simultanen) Zertifikat-Anfragen bewältigen. Ein Peer-to-Peer System bietet immanente Lastverteilung. Die Arbeitsbelastung wird auf die teilnehmenden Nodes verteilt. Moderne P2P-Netze sind dafür ausgelegt, mehrere Billionen Nutzer und ein Datenvolumen von über 10^{14} Dateien (Zertifikate) zu bewältigen (vgl. [SF02]).

3 Sicherheitsmechanismen

Die digitale Signatur eines Public-Key-Zertifikats sichert dessen Authentizität und Integrität. Somit eignen sich Zertifikate gut zur verteilten Speicherung auf nicht vertrauenswürdigen Nodes eines P2P-Netzes. Es sind aber Mechanismen einzuführen, welche die Verfügbarkeit von Zertifikaten absichern. Bösartige Nodes könnten Zertifikate zurückhalten oder löschen und auf diese Weise den PKI-Dienst beeinträchtigen. Das Ziel eines derartigen Angriffs ist nicht die Authentizität der öffentlichen Schlüssel sondern die Verfügbarkeit des PKI-Dienstes.

3.1 Replikation

Zertifikate werden nicht nur in einer einzigen Instanz, sondern in mehreren Replikaten abgelegt. Fällt eine Node aus oder löscht eine Node ein Replikat, ist wegen der verbleibenden Replikate das Zertifikat nicht verloren. Die möglichst gleichmäßige Verteilung der Replikate eines Zertifikats basiert auf einer Hashfunktion, die zur Adressierung eingesetzt wird (vgl. Abschnitt 3.3).

```
publish(certificate Cert(X, PX, Y, PY)){
    for i = 1 upto rpl
      symkey = h(Y|PY|i);
      enccert = E_symkey(Cert(X, PX, Y, PY));
      address = h(symkey);
      set(enccert, address);
    }
```

Abbildung 1: Veröffentlichungsprozess

```
retrieve_cert(subject Y, publicKey PY){
    for i = 1 upto rpl
      symkey = h(Y|PY|i);
      address = h(symkey);
      enccerts = get(address);
      certs = certs ∪ D_symkey(enccerts);
    return certs;
    }
```

Abbildung 2: Suchprozess

3.2 Rekursives Routing

Eine Anfrage für ein Zertifikat findet mit Hilfe mehrerer Zwischen-Nodes (Hops) ihr Ziel. So richtet zum Beispiel Node A ihre Anfrage an Node B, diese fragt Node C, usw., bis die Anfrage Node Z erreicht, welche das geforderte Zertifikat vorliegen hat. Für die Sicherheit der PKI ist entscheidend, dass die Antwort auf die Anfrage *nicht direkt* von Node Z an die ursprünglich anfragende Node A zurückgeschickt wird. Wäre das der Fall, könnte die Node A mittels einer einfachen Suchanfrage feststellen, dass Node Z für ein Replikat eines bestimmten Zertifikats zuständig ist. Diese Information würde Node A einen Denial-of-Service Angriff gegen die Zertifikate eines bestimmten Benutzers ermöglichen. Vielmehr muss die Antwort auf eine Anfrage den umgekehrten Weg wie die Anfrage selbst verwenden. Eine Node kann somit nicht in Erfahrung bringen, ob die Antwort unmittelbar von der gefragten Node stammt oder ob die gefragte Node die Antwort weiterreicht.

3.3 Verschlüsseltes Speicherschema

Durch die Verschlüsselung von Zertifikaten wird es einer bösartigen Node erschwert, den Inhalt eines Zertifikats im eigenen Speicherbereich oder beim Weiterreichen des Zertifikats zu interpretieren und somit bestimmte Zertifikate zurückzuhalten. Um das Verschlüsselungsschema mit Hilfe von Pseudocode zu beschreiben, wird folgende Notation eingeführt: $Cert(X, PX, Y, PY)$ bezeichnet ein Public-Key-Zertifikat, ausgestellt und digital signiert von Teilnehmer X. Die digitale Signatur lässt sich mit Hilfe des öffentlichen Schlüssels PX überprüfen. Das Zertifikat bescheinigt dem Teilnehmer Y den öffentlichen Schlüssel PY.

Die Verschlüsselung wird während des Veröffentlichungsprozesses (vgl. Abbildung 1) eines Zertifikats realisiert. Es werden rpl Replikate (systemweite Konstante) eines Zertifikats veröffentlicht. Der Prozess $publish$ bestimmt zunächst den symmetrischen kryptographischen Schlüssel für das Zertifikat $Cert(X, PX, Y, PY)$. Zu diesem Zweck wird die kollisionsresistente Einweg-Hashfunktion h verwendet. Der symmetrische Schlüssel wird berechnet als Hashwert der Konkatenation des Subjekt-Namens Y, des öffentlichen Schlüssels PY des Subjekts und der laufenden Replikat-Nummer i. Anschließend wird das Zertifikat mit Hilfe der Verschlüsselungsfunktion E_{symkey} verschlüsselt. Zur Adressierung des Zertifikats dient der Hashwert des symmetrischen Schlüssels: $address = h(symkey) = h(h(Y|PY|i))$. Dies hat zur Folge, dass jeder, der den symmetrischen Schlüssel eines Zertifikats kennt, das Zertifikat im P2P-Netz finden kann. Weiterhin kann

die Node, die das Zertifikat speichert und deswegen die Adresse des Zertifikats kennt, das Zertifikat nicht entschlüsseln, denn sie kann den symmetrischen Schlüssel nicht aus der Adresse rekonstruieren (Einweg-Eigenschaft der Hashfunktion h). Zuletzt wird das verschlüsselte Zertifikat durch die protokollspezifische Funktion set ins P2P-Netz eingestellt. Die Funktion set verwendet hierzu das oben angesprochene rekursive Routing.

Der Suchprozess für Zertifikate (vgl. Abbildung 2) realisiert die Entschlüsselung. Eine Node interessiert sich für die Zertifikate, die einem bestimmten Subjekt Y den öffentlichen Schlüssel PY bescheinigen. Gerade diese Information war (zusammen mit der laufenden Replikat-Nummer i) die Grundlage für die Adressierung und die Verschlüsselung. Zuerst wird basierend auf der Suchanfrage der symmetrische Schlüssel berechnet: $symkey = h(Y|PY|i)$. Darauf aufbauend wird die Adresse bestimmt: $address = h(symkey)$. Die protokollspezifische Funktion get liefert alle verschlüsselten Zertifikate, die unter dieser Adresse gespeichert sind. Abschließend werden diese Zertifikate mit Hilfe der Entschlüsselungsfunktion D_{symkey} entschlüsselt. Zuletzt liefert der Suchprozess alle gefundenen Zertifikate für Subjekt Y mit dem Schlüssel PY.

4 Fazit

Es wurden die Vorteile der Realisierung einer PKI auf Basis eines P2P-Netzwerks vorgestellt. Außerdem wurden Sicherheitsmechanismen eingeführt, welche die Verfügbarkeit von Zertifikaten in dieser nicht vertrauenswürdigen Umgebung absichern. Aktuelle Arbeiten beschäftigen sich mit der Erweiterung einer P2P-PKI um Gültigkeitsdauer für Zertifikate und Zertifikat-Rückruf. Ein weiteres Forschungsgebiet ist der Ausbau einer P2P basierten PKI zu einer Authentifizierungs- und Autorisierungsinfrastruktur (AAI).

Literatur

[Abe01] Karl Aberer. P-Grid: A Self-Organizing Access Structure for P2P Information Systems. In *Prodeedings of the Sixth International Conference on Cooperative Information Systems (CoopIS2001)*, number 2172 in Lecture Notes in Computer Science. Springer Verlag, 2001.

[AL03] Carlisle Adams und Steve Lloyd. *Understanding PKI - Second Edition*. Addison-Wesley, 2003.

[DHA03] Anwitaman Datta, Manfred Hauswirth und Karl Aberer. Beyond "web of trust": Enabling P2P E-commerce. In *Proceedings of the IEEE Conference on Electronic Commerce (CEC03)*, 2003.

[SF02] Detlef Schoder und Kai Fischbach. *Peer-to-Peer*. Springer, 2002.

[SMLN+03] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek und Hari Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications. In *IEEE/ACM Transactions on Networking*, Jgg. 11, 2003.

[Wöl05] Thomas Wölfl. Public-Key-Infrastructure Based on a Peer-to-Peer Network. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS38)*. IEEE Computer Society, 2005.