Zugriffskontrolle für Ortsinformationen in Nexus - Eine Gratwanderung zwischen Sicherheit und Funktionalität

Vergleich verschiedener Mechanismen zur Realisierung einer Zugriffskontrolle für Ortsinformationen in der Nexus-Plattform

SFB 627: Nexus - Umgebungsmodelle für Mobile Kontextbezogene Systeme
Universität Stuttgart
Arbeitsgruppe Berechtigungssysteme
Andreas Gutscher
gutscher@ikr.uni-stuttgart.de
23.1.2006

Zusammenfassung

In Plattformen für kontextbezogene Dienste werden Kontextinformationen, insbesondere auch Ortsinformationen von Benutzern, gespeichert und verarbeitet. Da sich aus Bewegungsmustern von Personen zum Teil sehr weitgehende Schlüsse über Tätgkeiten, Gewohnheiten und Vorlieben der Betroffenen extrahieren lassen, müssen personenbezogene Ortsinformationen als sehr sensible Daten angesehen und deshalb sowohl vor Dritten als auch vor den Betreibern der Plattform selbst geschützt werden. Da diese Anforderung mit herkömmlichen Zugriffskontrollmechanismen nicht erfüllt werden kann, wird in diesem Bericht der Einsatz von verschieden zusätzliche Schutzmechanismen am Beispiel der Nexus-Plattform dargestellt und bewertet.

1 Motivation

Nexus[1][1] ist eine Plattform für kontextbezogene Dienste, welche Kontextinformationen, insbesondere auch Ortsinformationen von Benutzern, speichert und verarbeitet. Es kann angenommen werden, dass viele Benutzer der Plattform grundsätzlich bereit wären, ihre Kontextinformationen in Nexus einzubringen, damit sie selbst und zum Teil auch andere Benutzer Dienste nutzen können, welche Zugriff auf diese Kontextinformationen benötigen. Benutzer werden einer Weitergabe ihrer Kontextinformationen nur dann zustimmen, wenn sie die Kontrolle über ihre Daten behalten und die Weitergabe nach ihren Wünschen einschränken können.

Da sich aus Bewegungsmustern von Personen zum Teil sehr weitgehende Schlüsse über Tätgkeiten, Gewohnheiten und Vorlieben der Betroffenen extrahieren lassen, müssen personenbezogene Ortsinformationen als sehr sensible Daten angesehen und entsprechend geschützt werden. Systeme, welche keinen sicheren, vom Benutzer konfigurierbaren

1. http://www.nexus.uni-stuttgart.de/

Zugriffsschutz bereitstellen, stellen eine starke Einschränkung der informationellen Selbstbestimmung dar, so dass deren Nutzung voraussichtlich von der überwiegenden Mehrheit der Benutzer als nicht akzeptabel gewertet werden wird.

2 Funktionale Anforderungen, Ortsinformationsabfragen

Der Location Service (LS) [2] kann den Standort (ev. auch vergangene Standorte) von Objekten speichern, die Einträge können von entsprechenden Sensoren (meist auf dem Endgerät des Benutzers) durch Location Updates aktualisiert und an anfragende Benutzer und Dienste herausgegeben werden. Der Location Sensoren soll hierbei die gewünschten Einschränkungen des Eigentümers beachten und durchsetzen. Der Event Service (ES) [3] ergänzt den LS um die Möglichkeit, Ereignisbeobachtungen auf den Daten des LS durchzuführen und den Anfragenden bei Eintreten des Ereignisses zu benachrichtigen.

Es existieren verschiedenen Anfragetypen, die wichtigsten sind:

- Object Queries:
  Anfrage nach dem Ort eines spezifizierten Objekts
  („Wo ist Alice in diesem Moment?“)
- Area Queries:
  Anfrage nach Objekten in einem spezifizierten Gebiet
  („Welche Institutsmitarbeiter befinden sich gerade im Gebäude V47?“)
- Events:
  An- und Abmeldnung von Benachrichtigungswünschen im Falle des Eintretens von bestimmten Bedingungen zwischen beliebigen Objekten.
  („Gib mir Bescheid, sobald Alice in meine Nähe (weniger als 100 m Abstand) kommt.“)

3 Sicherheits-Anforderungen

Eigentümern von Kontextinformationen soll es möglich sein, diese anderen Personen oder Diensten in dem Maße und mit den Einschränkungen zur Verfügung stellen zu können, wie sie es wünschen (Zugriffskontrolle).

Als „Eigentümer“ von Kontextinformationen ist diejenige Person zu sehen, die die „Hoheit“ oder Befugnis über das Objekt hat, bei Kontextinformationen von Personen ist das im Normalfall die Person selbst, bei Kontextinformationen von Gegenständen und Ereignissen der Besitzer bzw. der Halter, Betreiber oder Veranstalter (die Klarstellung der Eigentumsfrage muss letztendlich in irgend einer Form juristisch geregelt sein bzw. werden).

Der Eigentümer soll die Möglichkeit haben, Einschränkungen auf bestimmte Personen(gruppen), Zielgebiete, Abfragegebiete, Zeiträume, Orts-/Zeit-/Messwertauflösungen, usw. gegenüber anderen Benutzern sowie Betreibern von Diensten zu definieren. Es sollen geeignete technische Mechanismen entworfen werden, um diese Einschränkungen auch wirksam gegen alle Beteiligten (insbesondere auch gegen die Betreiber der Dienste) durchzusetzen.

4 Randbedingungen, Angreifermodell

Die Durchsetzung von Zugriffseinschränkungen gestaltet sich unterschiedlich schwierig, je nachdem, ob die beteiligten Betreiber als vertrauenswürdig (hinsichtlich ihrer Korrektheit, Zuverlässigkeit, Kompetenz, Gutwilligkeit, Durchsetzungsfähigkeit, usw.) eingestuft werden können.

Da Nexus im Allgemeinen ein weitgehend offenes System definiert, in dem eine sehr große Anzahl von Dienstbetreibern und Benutzern auftreten können, kann nicht immer davon ausgegangen werden, dass alle Benutzer alle Betreiber als voll vertrauenswürdig einstufen werden.

Es sind deshalb die folgenden Angreifermodelle zu nennen:

1. Alle Betreiber sind vertrauenswürdig, Schutz nur gegen andere Benutzer und Dritte erforderlich.
2. Manche der beteiligten Betreiber sind vertrauenswürdig, andere hingegen nicht (je nach Szenario).
3. Keiner der beteiligten Betreiber kann als vertreuenswürdig angenommen werden.

Sind alle beteiligten Betreiber vertrauenswürdig, dann kann eine Durchsetzung der Zugriffskontrolle relativ einfach realisiert werden. Je mehr Betreiber aber als nicht vertrauenswürdig eingestuft werden müssen, um so aufwendiger ist das Finden einer technischen Lösung. Die Anforderungen zur effektiven Durchsetzung von Zugriffskontrollen auch gegen Betreiber, welche dies Daten verarbeiten müssen, steht zum Teil in einem inherenten Konflikt mit der Anforderung einer uneingeschränkten performanten Verarbeitung der Daten und ist signifikant schwieriger zu realisieren, u.U. ist sogar unklar, ob überhaupt eine Lösung gefunden werden kann.

5 Problemanalyse

5.1 Schutz von Object Queries

Für den Zugriff über Object Queries kann ein Schutz relativ einfach realisiert werden, indem Kontextinformationen verschlüsselt auf dem LS abgelegt werden, berechtigten Benutzern und Diensten lässt man den zugehörigen Schlüssel über einen vertraulichen Kanal zukommen so dass alle Berechtigten die Kontextinformationen damit wieder entschlüsseln und lesen können.

5.2 Schutz von Area Queries und Events

Soll der LS die die Ortsinformationen auch in Area Queries und Events berücksichtigen, so ergibt sich folgendes Problem:

Um Area Queries effizient durchführen zu können, müssen alle Ortsinformationen in auswertbarer Form (also insbesondere nicht verschlüsselt) auf wenigen leistungsfähigen Datenbank-Servern des LS-Betreibers vorliegen. Die Eigentümer der Ortsinformationen möchten aber nicht, dass die LS-Betreiber ihre Daten im Klartext haben, da der LS im Allgemeinen nicht als hinreichend vertrauenswürdig angesehen wird.

6 Mechanismen

Im Folgenden werden bekannte und neu entworfene Mechanismen zur Lösung des Problems vorgestellt und bewertet hinsichtlich der Umsetzbarkeit, eventuellen Funktionseinschränkungen, der Sicherheit und dem verursachten Aufwand.

6.1 Eigener LS

Der Benutzer (als Eigentümer seiner eigenen Kontextinformation) betreibt hierzu einen eigenen, kleinen LS, der seine Kontextinformationen speichert, Anfragen nach diesen Kontextinformationen müssen dann direkt an diesen LS gerichtet werden. Dadurch muss sich der Benutzer nicht mehr auf einen LS-Betreiber verlassen, sondern er behält die volle Kontrolle über seine Daten bei sich.

Mit diesem eigenen LS sind Object Queries und im Prinzip auch Area Queries und Events möglich, bei letzteren entstehen aber sehr große Performance- und Skalierungprobleme. Für die Beantwortung von Area Queries sowie für die Auswertung vieler Eventtypen können die Ergebnisse nicht mehr wie bisher effizient innerhalb einer oder weniger Datenbanken ermittelt werden, stattdessen müssen sehr viele LS abgefragt werden, die nur über einen möglicherweise schmalbandigen und teueren Netzzugang sowie geringe Rechenleistung verfügen, so dass die Auswertungen hohe Verzögerungen aufweisen, teuer sind und die kleinen mobilen Endgeräte überlasten.

6.2 TPM

6.2.1 Funktionsweise

Die Dienste werden auf zertifizierter Hardware ausgeführt, welche mit einem Trusted Platform Module (TPM) [7] ausgestattet ist. Die Betreiber veröffentlichen die vollständige Hard- und Softwarekonfiguration der eingesetzten Systeme, so dass geprüft werden kann, dass die Systeme die Daten ohne Gefährdung oder Missbrauch von Daten abläuft.

Über Remote Attestation kann die Integrität der Hard- und Software überprüft werden.

6.2.2 Sicherheit

Wenn es tatsächlich gelingt, alle Hard-, Firm- und Software-Komponenten der Systeme eingehend zu prüfen und zu verifizieren, dass diese gemäß ihrer Spezifikation funktionieren und dass unberechtigte Zugriffe und Missbrauch der Daten sicher ausgeschlossen werden kann, dann kann das Verfahren als sicher angesehen werden. Es ist jedoch davon auszugehen, dass eine derart umfassende Prüfung im Regelfall nicht möglich sein wird und dass es immer mehr oder weniger einfach auffindbare Lücken in mindestens einer der komplexeren Komponenten geben wird. Ein Fehler in einer Komponente der Sicherheitskette genügt, um dem Schutz vollständig auszuhebeln.

6.2.3 Funktionale Einschränkungen

Betreiber müssen insbesondere die vollständige Softwarekonfiguration (Quelltexte) veröffentlichen und können keine kurzfristigen Änderungen vornehmen.

6.2.4 Mehraufwand

Im Betrieb wird voraussichtlich nur vernachlässigbarer Mehraufwand entstehen, die eingehende Prüfung von Hard-, Firm- und Software der Systeme hingegen wird einen enormen Arbeitsaufwand erfordern.

6.3 Pseudonymisierung

Die Kontextinformationen werden hierbei im Klartext auf dem LS gespeichert, so dass diese für Area Queries und Event-Anmeldungen verarbeitet werden können, man speichert sie aber nicht mit seinem bekannten Bezeichner sondern unter einem anderen (z.B. zufällig gewählten) Bezeichner (= Pseudonym), so dass die Vertraulichkeit dadurch geschützt wird, dass Unberechtigte zwar die Kontextinformationen lesen können, aber nicht wissen, zu welchem Objekt sie gehören [4]. Berechtigten Benutzern und Diensten kann der Besitzer sein Pseudonym mitteilen. Das Pseudonym kann ggf. häufig gewechselt werden, um eine Beobachtung des Pseudonyms über einen längeren Zeitraum zu verhindern, denkbar ist auch der Einsatz von Dummy-Pseudonymen.

Nachteilig ist erstens, dass Unberechtigte bei Gebietsabfragen durchaus sehen, dass und wo sich Objekte befinden, auch wenn sie diese nicht einem bestimmten Objekt zuordnen können und diese teilweise Dummy-Pseudonyme sein können. Zweitens kann der LS trotz Pseudonymisierung oftmals die Zusammengehörigkeit von Pseudonymen anhand der IP-Adresse des Objekts/Besitzers erkennen, sofern dies nicht durch zusätzliche Maßnahmen verhindert wird [5]. Drittens ist zu befürchten, dass eventuell sehr viele falsche Events von Dummy-Werten ausgelöst werden und dass Event Tracker durch Dummy-Werte gestört werden könnten (insb. durch Sprünge). Viertens können Events nicht auf wechselnde Pseudonymen angemeldet werden und es ist fünftens auch ein zusätzlicher Aufwand, das Pseudonym aufzulösen und zu verwalten sowie Echte von Dummywerten zu trennen.

6.4 Dummy-Daten

6.4.1 Funktionsweise

Neben den tatsächlichen Daten werden gezielt auch unzutreffende Daten (Dummy-Daten) in den LS abgelegt, so dass Angreifer nicht ohne Weiteres wissen können, welche Daten zutreffend sind.

Es muss eine Möglichkeit geben, die Daten durch Meta-Daten zu markieren, so dass Berechtigte Benutzer mit Hilfe von weiteren Informationen ermitteln können, ob es sich um Dummy-Daten handelt oder nicht. Die kann beispielsweise dadurch realisiert werden, dass alle Werte mit einer verschlüsselten Kennzeichnung versehen werden, so dass nur berechtigte Benutzer mit Hilfe der dazu benötigten Schlüssel diese Kennzeichnung entschlüsseln und prüfen können.

6.4.2 Funktionseinschränkungen

Es sind keine Funktionseinschränkungen zu erwarten, solange sichergestellt wird, dass immer ausreichend viele Daten auf den LS abrufbar sind, damit sichergestellt ist, dass mindestens einer dieser Werte kein Dummy-Wert ist.

6.4.3 Sicherheit

Relativ leicht angreifbar, da die tatsächlichen Daten ungeschützt vorliegen und z.B. durch Plausibilitätsprüfungen oder statistische Analysen Informationen über den tatsächlichen Ort oder über Bewegungsmuster herausgefiltert werden können.

6.4.4 Mehraufwand

Die Methode führt zu einer Vervielfachung des Datenbestandes, dadurch entsteht ein erhöhter Speicherbedarf in den Datenbanken, die Suche wird entsprechend aufwendiger, es werden viele unzutreffende Ergebnisse auf Anfragen übermittelt und viele unzutreffende Events ausgelöst, die übertragen werden müssen, bevor sie vom Benutzer überprüft und gefiltert werden können.

6.5 Verschlüsselung der Ortsdaten

6.5.1 Funktionsweise

Ortsdaten werden (symmetrisch) verschlüsselt auf dem LS gespeichert. Es wird markiert, mit welchem Schlüssel die Daten verschlüsselt sind, berechtigte Benutzer erhalten einen Schlüssel zum Entschlüsseln der Daten.

6.5.2 Funktionseinschränkungen

Events können entweder gar nicht mehr oder nur noch von vertrauenswürdigen Komponenten (z.B. von einem Agenten oder dem mobilen Endgerät) ausgewertet werden. Eventuell können Events nicht oder erst zu spät erkannt werden, da die Daten erst zu einer Komponente transportiert werden müssen, an der sie entschlüsselt werden können.

6.5.3 Sicherheit

Die Verschlüsselung bietet ein hohes Maß an Sicherheit, Angriffe können relativ zuverlässig verhindert werden.

6.5.4 Mehraufwand

Es entsteht nur geringer Mehraufwand an Rechenleistung durch die Ver- und Entschlüsselung von Daten, es entsteht jedoch ein hoher Mehraufwand bezüglich der Kommunikation durch die ineffiziente lokale Eventauswertung.

6.6 Transformation von Ortsdaten

Bei diesem Verfahren werden Koordinaten in ein anderes Koordinatensystem transformiert, welches gegenüber dem natürlichen verschoben und gedreht sein kann. Berechtigte Benutzer erhalten die Transformationsfunktion, mit der sie wieder die ursprünglichen Werte berechnen können, Nichtberechtigte sehen hingegen nur falsche Daten. Events können angemeldet werden, wenn alle Punkte dabei die gleiche Transformation verwenden (z.B. muss ein Area-Event auf das transformierte Gebiet angemeldet werden). Das Verfahren ist in [6] ausführlich dargestellt.

Nachteilig ist, dass sich aus Bewegungsprofilen dennoch Informationen ableiten lassen, auch wenn sie transformiert sind (eine Person befindet sich zu Hause, hat einen festen Arbeitsplatz, ...), eventuell lässt sich die Transformation nach der Auswertung von gesammelten Orts-Historien ermitteln (z.B. durch Suche nach Übereinstimmungen mit der Topologie von Verkehrswegen), zudem ist auch eine Verwaltung der Transformationen notwendig.

6.6.1 Sicherheit

Es bestehen gewisse Risiken, dass durch die Verkettung von Differenztransformationen Transformationsfunktionen möglicherweise rekonstruiert werden können, Untersuchungen hierzu sind jedoch noch nicht abgeschlossen.

7 Kombinationen der Mechanismen

Der Einsatz von TPMs kann mit allen anderen Schutzmechanismen kombiniert werden, ebenso der Einsatz von VIDs.

Die Ortsdaten können entweder verschlüsselt oder transformiert werden (beides gleichzeitig ist nicht sinnvoll), zusätzlich können Dummywerten eingesetzt werden.

8 Fazit

8.1 Bewertung, allgemein

Ein sicherer Schutz der Ortsinformationen bei Object Queries ist relativ einfach lösbar, bei Area Queries und Events hingegen ist kein optimaler Schutzmechanismus bekannt. Die vorgestellten Lösungsansätze sowie Kombinationen davon verursachen einen relativ hohen Aufwand und weisen zum Teil geringe Funktionseinschränkungen auf. Die Schutzmechanismen sind weitgehend miteinander kombinierbar bzw. parallel einsetzbar, so dass durch die Auswahl bzw. Kombination eine gewisse Abwägung zwischen Sicherheit und Funktionalität und Aufwand möglich ist.

8.2 Konzept für Nexus

Die Dummy-Methode ist relativ unsicher und aufwendig und sollte nicht eingesetzt werden. Der Einsatz von TPMs erscheint derzeit unrealistisch und wäre aus Sicht der Dienste und Anwendungen transparent.

Die andere Mechanismen und Kombinationen davon (soweit möglich) können jedoch eingesetzt werden:

VIDs können und sollten in jedem Falle ergänzend zu allen anderen Mechanismen eingesetzt werden.

Anwender mit hohen Sicherheitsanforderungen sollten die Ortsdaten nur verschlüsselt auf dem LS ablegen. Anwender, welche den Event Service uneingeschränkt nutzen möchten und nur mittleren Sicherheitsanforderungen haben, können die Transformationsmethode

einsetzen. Bei geringen Sicherheitsanforderungen kann unter Umständen sogar der Einsatz von VIDs ausreichen.

Die Wahl der Methode sollte nach Möglichkeit dem Benutzer überlassen werden, Betreiber sollten nach Möglichkeit alle Kombinationen unterstützen.

Literatur

[1] D. Nicklas and B. Mitschang. On building location aware applications using an open platform based on the NEXUS Augmented World Model. In Software and Systems Modeling, 2004.

[2] A. Leonhardi and K. Rothermel. Architecture of a Large-scale Location Service. Universität Stuttgart, Institut für Parallele und Verteilte Höchstleistungsrechner, Fakultätsbericht Nr. 2001/01, 2001.

[3] M. Bauer. Event Management for Mobile Users. Universität Stuttgart, Institut für Parallele und Verteilte Höchstleistungsrechner, Fakultätsbericht Nr. 2004/02, 2004.

[4] C. Hauser and M. Kabatnik. Towards Privacy Support in a Global Location Service. In Proceedings of the IFIP Workshop on IP and ATM Traffic Management (WATM/ EUNICE), 2001.

[5] C. Hauser. Mobility Management Meets Privacy — The Failure of Existing Proposals and a New, Future-Proof Approach. In Proceedings of the Second International Workshop on Mobility Management & Wireless Access Protocols, 2004.

[6] A. Gutscher. Coordinate Transformation - A Solution for the Privacy Problem of Location Based Services? To appear in Proceedings of the 2nd International Workshop on Security in Systems and Networks, 2006.

[7] Trusted Computing Group, `https://www.trustedcomputinggroup.org/`