

Entwicklung eines Proxy-Servers für eine transparente, digitale Signierung von E-Learning-Inhalten

Dipl.-Inf. Christian J. Eibl
Dipl.-Inf. Kirstin Schwidrowski

20. April 2006

1 Organisatorisches

Thema:	Entwicklung eines Proxy-Servers für eine transparente, digitale Signierung von E-Learning-Inhalten
Zeitraum:	voraussichtlich Mai 2006 bis Mai 2007
Teilnehmerzahl:	ca. 6-12
Veranstalter:	Christian J. Eibl (eibl@die.informatik.uni-siegen.de) Raum: H-A 7103, Tel.: 0271/740-3233

2 Teilnahmevoraussetzungen

Für die Entwicklung eines Proxy-Servers werden gute Programmierkenntnisse in Java vorausgesetzt [1, 2]. Weiterhin werden Kenntnisse der Software-Technik erwartet, um den Entwicklungsprozess der Software effizient und qualitativ hochwertig zu gestalten. Die technischen Hintergründe für die Netzwerkprogrammierung und digitale Signaturen erfordern darüber hinaus Kenntnisse von Rechnernetzen und deren Funktionsweise, sowie grundlegende mathematische und kryptographische Kenntnisse [3].

Zusammengefasst sollen folgende Inhalte bekannt sein:

- Java-Programmierung,
- Software-Technik,
- Rechnernetze,
- ggf. Kryptographie.

3 Motivation

3.1 Hintergrund

Ein *Proxy-Server*, kurz *Proxy*, ist eine Anwendung, die wortgemäß dem Benutzer nahe steht. Sie stellt eine Art Vermittler dar, der statt dem eigentlichen Anwender mit einer entfernten Anwendung bzw. einem entfernten Server kommuniziert. Der Proxy arbeitet netzwerktechnisch auf der Anwendungsschicht und ist damit in der Lage, alle Inhalte, die vom Benutzer gesendet werden bzw. an ihn gerichtet sind, zu

lesen und zu verändern. Aus diesem Grund ist es möglich, den Datenverkehr zwischen Client (Lehrender/Lernender) und Server (Lern-Management-System, kurz LMS) zu überprüfen und diesen mit Hilfe kryptographischer Methoden zu sichern. Die Form von Kryptographie, die hierbei verwendet werden soll, ist die digitale Signatur. Die digitale Signatur besteht aus einem Hash-Wert, also einer Art Fingerabdruck der zu sendenden Nachricht, der mit Hilfe eines privaten Schlüssels verschlüsselt und der Klartextnachricht angehängt wird. Es finden hier also sowohl Hash-Funktionen als auch asymmetrische Kryptographie Anwendung. Über die Verbreitung des zum privaten Schlüssel passenden öffentlichen Schlüssels und der Mitteilung, welche Hash-Funktion verwendet wurde, kann die empfangene Nachricht mit der angehängten Signatur geprüft werden. Hierzu wird der öffentliche Schlüssel verwendet, um den Hash-Wert zu entschlüsseln. Die Hash-Funktion hingegen berechnet einen eigenen Wert zu der empfangenen Klartextnachricht. Sind der erhaltene Hash-Wert und der eigens berechnete Hash-Wert gleich, so kann eine Manipulation der Nachricht mit hoher Wahrscheinlichkeit ausgeschlossen werden. Im Falle eines anderen Ergebnisses gilt eine Manipulation als nachgewiesen.

3.2 Fachliche Begründung für das Projekt

Bei Betrachtung von Lern-Management-Systemen, wie sie in der Praxis eingesetzt werden, stellt sich schnell heraus, dass diese Software häufig nur notdürftig abgesichert ist. Am Beispiel des Lern-Management-Systems Moodle, das u.a. auch an der Universität Siegen zur Unterstützung der Lehre eingesetzt wird, bedeutet dies, dass Zugriffe auf vertrauliche Daten und Manipulationen von Lerninhalten und Kommunikationsnachrichten nur durch den Prozess der Authentifizierung und der damit verbundenen rollenbasierten Zugriffskontrolle geschützt sind. Sofern es einem Angreifer gelingt, die Zugangsdaten eines Teilnehmers, sei es Lehrender oder Lernender, zu stehlen, so kann er gemäß der Berechtigung dieser Person unbemerkt mit dem System interagieren und Inhalte verändern. Um eine zusätzliche Hürde bei solchen Identitätsdiebstählen zu etablieren, lassen sich digitale Signaturen nutzen, die jeder Nachricht von Seiten des Benutzers angefügt werden. Mit anderen Worten, es wird jede noch so kurze Nachricht an das System vom Nutzer unterschrieben. Im Fall von Identitätsdiebstahl kann dann der Angreifer zwar noch vertrauliche Dokumente betrachten, jedoch nicht mehr unbemerkt Änderungen vornehmen, da der Zugriffsschutz durch eine zusätzliche digitale Signatur ergänzt wurde und Manipulationen sofort offenkundig würden. Hier wird implizit angenommen, dass der Angreifer nicht auch noch den privaten Schlüssel der Person kennt, deren Identität er aktuell verwendet!

Die digitale Signierung jeder Nachricht, die das System verlässt, kann jedoch vor allem bei sehr kurzen und vermeintlich unwichtigen Nachrichten die Geduld des Benutzers stark strapazieren. Der Benutzer müsste das Verfahren aus dem vorherigen Unterabschnitt für jede noch so kurze Nachricht komplett durchlaufen, um das Risiko unbemerkter Manipulation zu umgehen. Das ist vor allem bei schnellen Kommunikationsmitteln mit kurzen Nachrichten, wie z.B. Chats, ein Aufwand, der nicht zu vertreten wäre. Aus diesem Grund soll eine automatisierte Herangehensweise implementiert werden, bei der die Signierung transparent für den Benutzer im Hintergrund stattfindet. Die Sicherheit könnte damit weiterhin bestehen, der Aufwand würde jedoch vom Benutzer auf das System übertragen.

Vorteil eines Proxy-Servers für diese Aufgabe besteht darin, dass bei einem Proxy auf Client-Seite der schützenswerte private Schlüssel den Rechner des Clients nicht verlassen müsste. Da die Browser und sonstigen Programme für die Kommunikation mit E-Learning Systemen in der Regel keine Mechanismen für eine automatisierte Signatur mitbringen, muss dieser Mechanismus ausgelagert werden. Weiterer Vorteil von Proxys ist, dass die meisten netzwerkgestützten Anwendungen in der Lage

sind, ihre Verbindungen über einen Proxy-Server herzustellen.

4 Aufgabenbeschreibung

In der Projektgruppe soll ein Proxy-Server implementiert werden, der auf dem Rechner eines Lernenden bzw. Lehrenden läuft und automatisch im Hintergrund alle Nachrichten an das LMS digital signiert. Eingehende Nachrichten vom LMS werden analog zu der Signierung mit Hilfe der bekannten öffentlichen Schlüssel bzgl. einer Manipulation überprüft. Der Benutzer ist im Fall einer festgestellten Manipulation über diesen Sicherheitsvorfall zu informieren.

4.1 Beschreibung des Proxys

Der zu erstellende Proxy-Server ist über die Einbindung einer extern zu speichernden Konfigurationsdatei variabel einzustellen auf die aktuelle Client-Server-Situation im Netzwerk, d.h. zu überwachender Port, IP-Adresse des LMS, Algorithmus zur Verarbeitung der Daten, u.v.m.

Durch eine möglichst stark forcierte Modularisierung und speziell durch die Erweiterbarkeit mit Hilfe dynamisch verlinkter Bibliotheken besteht die Möglichkeit, die Funktionalität des Proxy-Servers ohne erneutes Übersetzen auszubauen. Es ergeben sich damit ein Basisprogramm, das die grundlegenden Funktionalitäten zur Verfügung stellt, sowie Module/Bibliotheken, die spezielle Funktionen für bestimmte Aufgaben bzw. Spezifikationen für Vorgehensweisen in solchen Situationen für das Basisprogramm liefern.

Auf keinen Fall darf die grundsätzliche Funktionalität des Netzwerkes aus Sicht des Benutzers verlorengehen, wenn der Datenverkehr über den zu erstellenden Proxy geleitet wird. Verbindungen, die auf Grund fehlender Information nicht durch den Proxy verarbeitet werden können, sind unverändert und ohne weitere Verzögerung weiterzuleiten.

Da E-Learning auch international Verwendung findet, benötigt der Proxy Internationalisierungsmöglichkeiten, z.B. Spracherweiterungen durch Sprachpakete, die nicht statisch mit dem Basisprogramm gekoppelt sind.

Der Proxy muss in der Lage sein, mit dem Benutzer auf graphischer Ebene zu interagieren. Die Konfiguration der Proxy-Einstellungen ist graphisch vorzunehmen und darf keine weiteren Vorkenntnisse von dem Benutzer erwarten.

4.2 Minimalziele

Als Minimalziel wird ein funktionstüchtiger Proxy-Server erwartet, der in Java mit Hilfe der Java-Netz-Bibliotheken (`java.net.*`) erstellt wird. Dieser Proxy ist mit Hilfe der Methoden der Software-Technik vor der Implementierung bzgl. seiner Anforderungen zu analysieren und ein geeigneter Entwurf zu erstellen. Während der Analysephase ist ein Pflichtenheft zu erarbeiten, anhand dessen das weitere Vorgehen diskutiert werden kann.

Bei der Implementierung ist zu beachten, dass der Proxy-Server einfach erweiterbar und konfigurierbar sein soll. Das bedeutet, dass die Anwendung über dynamisch zu ladende Bibliotheken an verschiedene Protokolle und LMS anpassbar sein soll. Da im einfachsten Fall jede Verbindung vom Benutzer zum Internet über diesen Proxy läuft, muss sichergestellt sein, dass Verbindungen, die nicht über eine dynamisch eingebundene Bibliothek verarbeitet werden können, ungehindert und unverändert weitergeleitet werden. Für diese Zwecke sind Threads zu verwenden, um den Proxy bei eingerichteter Verbindung für weitere Verbindungen nicht zu blockieren.

Bezüglich der Internationalisierbarkeit sind Sprachpakete dynamisch einzubinden

für die gewählte Sprache. Als Minimalziel ist hierbei die Unterstützung von Deutsch und Englisch nötig.

Für den Fall, dass sich der Benutzer in einem Netzwerk befindet, das nur über einen Proxy-Server Zugang zum Internet hat, muss unser Proxy-Server auf variablen Ports lauffähig sein und die Verbindung nach eigener Verarbeitung an einen anderen Proxy weiterreichen können. Es muss also auch in einer Kette von Proxy-Servern die Funktionalität erhalten bleiben.

Die Anforderungen lassen sich der folgenden Aufzählung entnehmen:

- Erstellung eines Pflichtenheftes zur Vorstellung von Lösungsansätzen,
- Entwurf mit softwaretechnischen Methoden,
- Verwendung von Threads, um mehrere Verbindungen zu unterstützen,
- Verwendung eines Datenpuffers (FIFO) mit schneller Weiterleitung an den Zielservers,
- Dynamisch zu ladende Module für verschiedene Einsatzzwecke/LMS
→ primär: Modul zur Verwendung mit dem LMS Moodle,
- Unterstützung für Internationalisierung,
- Rapid Prototyping zur frühzeitigen Erkennung eventueller Entwurfsfehler,
- Plattformunabhängigkeit (Programmierung in Java),
- Kommunikationsmöglichkeit mit dem Benutzer über Web-Interface, ggf. Veränderung des Datenstroms zur sofortigen Meldung (in dem Fall muss richtiger Strom gespeichert werden),
- hohes Maß an Konfigurierbarkeit,
- Unterstützung von Proxy-Ketten, d.h. Weiterleitung des Datenstroms an weitere Proxy-Server.

Die Konfiguration des Proxy-Servers, sowie die Kommunikation mit dem Benutzer kann über ein Web-Interface erfolgen, da in der Regel bereits ein Web-Browser durch den Benutzer geöffnet ist. Alternativ kann abhängig vom Vorankommen der Projektgruppe die graphische Konfiguration durch ein vorläufiges, manuelles Anpassen der Konfigurationsdateien ersetzt werden.

4.3 Erweiterungen

Es ergeben sich zahlreiche Möglichkeiten, den Proxy wie oben beschrieben zu erweitern. Prinzipiell muss hierbei unterschieden werden, ob sich Erweiterungen auf das Basisprogramm oder die Erstellung von Zusatzmodulen beziehen.

Als wichtigste, wenn auch im Minimalziel auf Grund der Komplexität nicht mehr zu erwartenden Leistung, ist die Funktionalität der vertraulichen Übermittlung von Daten zu sehen. Da der Proxy bei einer verschlüsselten Verbindung (mittels SSL/TLS) zwischen Benutzer und Server keine Eingriffe vornehmen kann, ist die Verschlüsselung zwischen dem Proxy und dem Server aufzubauen. Hierzu müssen SSL/TLS-Komponenten implementiert werden, die gemäß dem Protokoll Verschlüsselungsverfahren aushandeln und den Verkehr entsprechend dieses Verfahrens und dem ausgehandelten Schlüssel fortan verschlüsseln. Die Erweiterung muss sich dabei an die Standards SSL der Versionen 1.0 bis 3.0 halten, sowie TLS 1.0 unterstützen. Diese Erweiterung ist die wichtigste der in diesem Abschnitt aufgeführten Erweiterungen des Proxys.

Weitere Möglichkeiten zur Erweiterung bestehen in der Programmierung von zusätzlichen Modulen, um andere LMS als Moodle zu unterstützen. Außerdem besteht die Möglichkeit bei einem Wechsel des Protokolls auch E-Mails durch den Proxy signieren zu lassen. Hierfür kann ebenfalls ein Modul implementiert und eingebunden werden.

Um einem Benutzer, der zwar Anwendungserfahrung mitbringt, jedoch unerfahren in der Java-Programmierung ist, die Möglichkeit zu geben, den Proxy selbstständig an ein neues LMS anzupassen, kann eine Erweiterung für den Proxy implementiert werden, die mit Hilfe von Benutzereingaben die nötigen Details erfragt, um mit einem bis dato unbekanntem LMS in der gewohnten Weise zu interagieren bei gleichzeitiger Unterstützung von digitalen Signaturen. Hierfür ist also eine Art Assistent zu erstellen, der die nötigen Fakten sammelt und analog zu einem Modul diese Konfigurationen speichert und zur Anwendung bringt.

5 Arbeitsorganisation

5.1 Arbeitsphasen

Während der Arbeit der Projektgruppe sind verschiedene Arbeitsphasen durchzuführen. Die folgende Aufzählung skizziert kurz diese Phasen und die dazugehörigen Arbeitsschritte:

1. **Analyse und Zeitmanagement**

In dieser Phase ist die Problemstellung zu prüfen und der zu erwartende Arbeitsaufwand zu analysieren. Anhand der fertigen Analysen ist eine Zeitaufstellung zu erarbeiten, die widerspiegelt, wie viel Zeit welcher Teilaufgabe zugedacht werden muss. Weiterhin soll ein Zeitplan erstellt werden, der wiedergibt, wann welche Aufgabe/Arbeitsphase spätestens begonnen und abgeschlossen sein soll.

2. **Erstellung eines Pflichtenheftes**

Verfeinerung der oben beschriebenen Anforderungen/Minimalziele und Anbringung von Lösungsansätzen für eine detaillierte Aufstellung, wie alle geforderten Problemfelder gelöst werden können. Dieses Pflichtenheft bildet die Basis für die weitere Arbeit mit aufgeteilten Aufgabenfeldern. Es stellt weiterhin die Grundlage für die Zwischenverteidigung dar, die in Form eines Vortrages vor den Mitgliedern der Fachgruppe zu erfolgen hat.

3. **Zuweisung von Teilaufgaben**

Nachdem die Schnittstellen und allgemeinen Diskussionen mit dem Pflichtenheft vorerst abgeschlossen sind, soll hier parallel an verschiedenen Teilaspekten gearbeitet werden.

4. **Implementierung, Test, Entwurfüberarbeitung**

Die Teilbereiche sind in Einzelarbeiten zu implementieren und in regelmäßigen Abständen zu Zwischentests zu verbinden. Erfahrungen und Ergebnisse aus diesen Testphasen sind ggf. im Entwurf anzupassen. Ein frühzeitig zur Verfügung stehender Prototyp ist wünschenswert, um ggf. grundsätzliche Entwurfsfehler bereits in einer frühen Phase erkennen zu können.

5. **Dokumentation, Projektbericht, Präsentation**

Alle Arbeiten sind ausreichend zu dokumentieren und in Form eines umfassenden Projektberichtes bei Abschluss der Arbeiten zu übergeben. Die Projektgruppe endet mit Erfüllen der Minimalziele und einer Abschlusspräsentation, die die Ergebnisse und Fakten aus der Zeit der Bearbeitung wiedergibt.

Im Gegensatz zu Praktika, in denen teilweise auch Teamarbeit Verwendung findet, ist in der Projektgruppe das Projektmanagement von großer Bedeutung.

5.2 Projektmanagement

Die Projektgruppenarbeit dient neben der Entwicklung der oben beschriebenen Software der Förderung der Teamfähigkeit der teilnehmenden Studierenden. Es wird ein hohes Maß an Kommunikationsbereitschaft innerhalb der Gruppe gefordert. Hierdurch wird der Informationsfluss innerhalb der Gruppe verbessert, so dass alle Teilnehmer zumindest grob über Arbeiten anderer Teilnehmer informiert sind und folglich ggf. auch Hilfestellung leisten können. Weiterhin besteht dadurch die Möglichkeit ggf. Arbeiten von evtl. ausgefallenen Teilnehmern zeitweise oder auf Dauer unkompliziert zu übernehmen. Eine hohe Kommunikationsbereitschaft resultiert damit in mehr Flexibilität bei Ausfällen und allgemein in gesteigerter Effizienz, was in Konsequenz zu einer erheblichen Zeitersparnis und besserer Motivation führt. In der folgenden Aufzählung werden die Erwartungen an das Projektmanagement kurz skizziert:

- Benennung eines Koordinators; späterer Wechsel nicht vorgesehen, jedoch ggf. möglich,
- regelmäßige Besprechungstermine; ca. wöchentlich nach Absprache,
- Protokollierung der Ergebnisse von Besprechungen und Diskussionen,
- Erstellung eines Zeitplans,
- Vorschlag zur zeitlichen und sachlichen Strukturierung,
- evtl. Vorschlag zu Arbeitsmethoden in den einzelnen Phasen (Einzel-, Teamarbeit, ...),
- Führen einer Übersicht der geleisteten Stunden (pro Person),
- Erarbeiten eines Abschlussberichtes unter Zuhilfenahme aller Protokolle und Ergebnisse, Darstellung der Aufgabe und der aufgetretenen Probleme, eingeschlagene Lösungswege mit Begründung, Diskussion der Ergebnisse.

Literatur

- [1] Ullenboom, C.: *Java ist auch eine Insel: Programmieren für die Java 2-Plattform in der Version 5*, 5. Auflage, Galileo Computing, 2006, URL: http://download.galileo-press.de/openbook/javainsel5/galileocomputing_javainsel5.zip [11.04.2006]
- [2] Boger, M.: *Java in verteilten Systemen: Nebenläufigkeit, Verteilung, Persistenz*, dpunkt Verlag, Heidelberg, 1999
- [3] Schneier, B.: *Angewandte Kryptographie*, Pearson Studium, München, 2006