Kryptographische Protokolle SSL/TLS und Web Services

Foued Jaibi
Seminar Innovative Internet-Technologien und Mobilkommunikation, WS 2008/2009
Institut für Informatik, Lehrstuhl Netzarchitekturen und Netzdienste
Technische Universität München
jaibi@in.tum.de

ABSTRACT (Kurzfassung)
Die Idee dieser Arbeit ist es, die Grundlagen von Web Sevices, kryptographischen Protokollen SSL/TLS und entsprechende Anwendungen zu charakterisieren und zu bewerten.

Keywords (Schlüsselworte)
SSL/TLS, Rekord protocol, Handshake protocol, Web Services, Security.

1. EINFÜHRUNG
In der Vergangenheit wurden Verschlüsselungstechniken fast nur im militärischen oder diplomatischen Bereich zur Sicherung und Geheimhaltung der Kommunikation genutzt. In der heutigen Zeit hat der „Information“ Begriff eine grosse Bedeutung für die Wirtschaft, insbesondere für deren verschiedene Bereiche wie Medizin, Industrie oder Dienstleistungen erlangt. Die Information ist zu einem unverzichtbaren Produktionsfaktor geworden und stellt sich vor allem als ein Wirtschaftsgut wie jedes andere absatzstarke Produkt dar. Dies ist ein deutlicher Grund dafür, dass gerade im Bereich der IT-Sicherheit intensiv geforscht wird.

2. SSL/TLS

2.1 Geschichte von SSL/TLS
Das Secure Sockets Layer SSL Protokoll wurde von der Firma Netscape entwickelt und zuerst als Version 2.0 im Jahre 1994 publiziert [8]. Netscape wollte damals ihren neuen und kryptographiefähigen Webserver besser auf dem Markt absetzen, indem sie einen freien Client - den „Netscape Navigator“ - zur Verfügung stellten. Dieser Web Browser hatte sicherlich auch die gleichen kryptographischen Protokolle unterstützt wie der entwickeleten Server. Seitdem erschienen auf dem Markt verschiedene neue Spezifikationen. Beispielsweise hat Microsoft ein ähnliches Protokoll, das PCT 1.0 im Jahr 1995 gebracht. Dies wurde in der ersten Version von Internet Explorer integriert. PCT 1.0 hat gegenüber SSL 2.0 einige Vorteile:“Die Umlauf und die Nachrichtenstruktur waren beträchtlich kürzer und einfacher“ [9]. Die Internet Engineering Task Force IETF entwickelte im Jahr 1999 auf Basis von SSL den Standard Transport Layer Security TLS. Im August 2008 erschien mit RFC 5246 die Version 1.2 von TLS, welche somit RFC 4346 (http://www.ietf.org) ersetzt hat. Als wesentliche Änderung steht jetzt keine direkte Abhängigkeit zwischen den Pseudozufallsfunktionen und den MD5/SHA-1 Algorithmen. Stattdessen wurde die Pseudorandom-Funktion neu definiert ([10],[11]).

2.2 Positionierung im Protokollstack
SSL wurde hauptsächlich entwickelt um sichere Internet Verbindungen zu ermöglichen. Es hat als Ziel, Information vor Manipulation, Missbrauch und Zugriff durch einen Unbefugten zu schützen. SSL ist oberhalb der Transportschicht (Bsp. TCP) und unter der Anwendungsschicht (Bsp. HTTP) angesiedelt und ist somit für andere Netzwerkprotokolle geeignet (siehe Abbildung 1) Man trifft besonders beim „Online Banking“ auf Seiten mit „https“. Dies ist nicht anders als in der Spezifikation beschriebene „HTTP over SSL“ Variante [6].

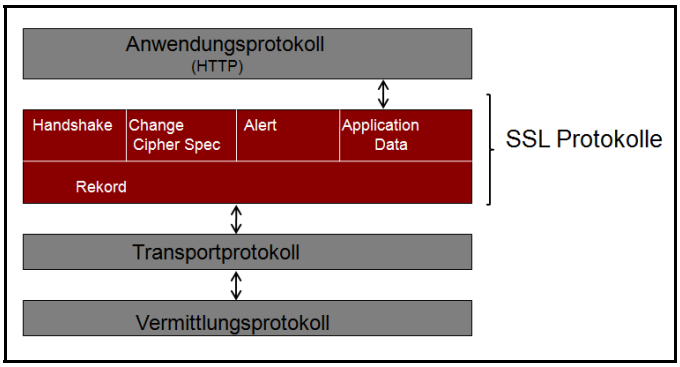

Abbildung 1: SSL im Protokollstack

2.3 Technische Ansatz

2.3.1 Das Rekord Protokoll
Das Rekord Protokoll stellt die untere Ebene des SSL Protokolls dar. Es stellt die Vertraulichkeit und Integrität von Nachrichten als Sicherheitsdienste bereit. Hierbei werden die zu übertragenden Anwendungsdaten in einzelene Pakete fragmentiert und danach kompremiert. In der Spezifikation von SSL/TLS [10] findet man keine explizite Beschreibung zur Kompressionsmethoden. Allerdings wird dort erwähnt, dass die Kompression keinen Verlust der Daten verursachen soll. Bei kleinere Datenmengen kann es zu einer Vergrösserung der Länge kommen. Hier darf die Länge der Daten nicht um mehr als 1024 Bytes vergrössert werden.
Zur Sicherung der Vertraulichkeit werden die zu übertragenden Daten verschlüsselt. Hierzu wird ein geheimer Schlüssel zwischen Client und Server vereinbart. Im Vergleich dazu wird zur Sicherung der Integrität ein MAC (Message Authentification Code) berechnet. Die genaue Funktionsweise vom Rekord Protokoll kann man in der folgenden Abbildung veranschaulichen:

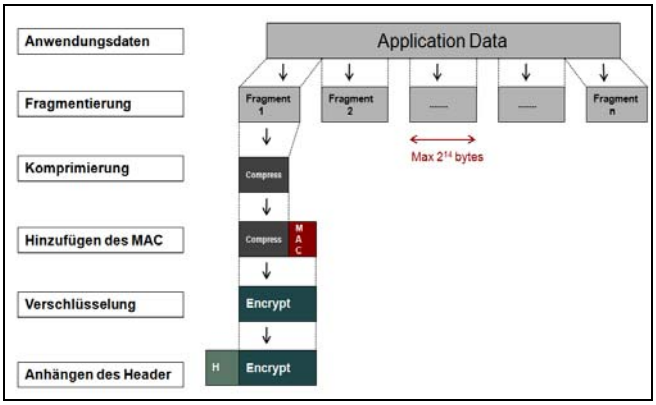

Abbildung 2: Funkionsweise des Rekordprotokolls

2.3.2 Das Handshake Protokoll

Durch das Handshake Protokoll wird zwischen dem Server und dem Client der Modus der Verschlüsselung, die Art der Nachrichtenauthentifizierung, der Schlüssel und alle zur Sicherung der Kanäle notwendigen Details vereinbart. Dieses Protokoll kann daher in vier wichtigen Phasen unterteilt werden [11]:

Phase 1: Festlegung der Ressourcen

Der Client schickt zum Server ein Client_Hello, woraufhin der Server mit einem Server_hello antwortet. Client_Hello kann dabei auch die Antwort auf ein Hello_request sein. Beim Nachrichtenaustausch müssen folgende Parameter bestimmt werden: die Version, eine Zufallzahl, eine Session ID und die zu verwendeten Cipher Suite.

In einer bestehenden SSL-Verbindung sorgt diese Phase außerdem für eine Neuverhandlung der Sicherheitsparameter.

Phase 2: Server Authentifizierung (optional)

Der Server identifiziert sich gegenüber dem Client. Hier wird auch das X509v3-Zertifikat [12] zum Client übermittelt. Außerdem kann der Server ein CertificateRequest an den Client schicken.

Phase 3: Client Authentifizierung(optional)

Hier identifiziert sich der Client gegenüber dem Server. Besitzt der Client kein Zertifikat, so antwortet er mit einem „NoCertificateAlert“. Der Client versucht außerdem, das Zertifikat, das er vom Server erhalten hat, zu verifizieren. Bei Misserfolg wird die Verbindung abgebrochen. Dieses Zertifikat enthält den öffentlichen Schlüssel des Servers. Wird die Cipher-Suite RSA verwenden, so wird das vom Client generierte pre-master-secret mit diesem öffentlichen Schlüssel verschlüsselt und kann vom Server mit dem nur ihm bekannten privaten Schlüssel wieder entschlüsselt werden. Alternativ kann hier auch das Diffie-Hellman-Verfahren[13] verwendet werden, um ein gemeinsames pre-master-secret zu generieren.

Phase 4: Beendigung des Handshake

Hier wird der Handshake beendet. Die change_cipher_spec veranlasst den Server, die gerade ausgehandelten Parameter für die weitere Sitzung zu übernehmen. „finished“ wird dann mit neuen Parametern verarbeitet. Der Server bestätigt mit „change_cipher_spec“ und „finished“.

2.3.3 Das ChangeCipherSpec Protokoll

Das ChangeCipherSpec Protokoll wird benutzt um von einem Verschlüsselungsalgorithmus zu einem anderen zu wechseln. Client und Server verhandeln über eine neue *CipherSpec* und einen neuen *Schlüssel*. Jede Entität schickt eine *CipherSpec* Nachricht, die den Beginn des Kommunikationsprozesses mit neuer *CipherSpec* und neuem *Schlüssel* veranlässt. Im Normalfall ändert sich das CipherSpec am Ende des SSL/TLS handshake. Allerdings kann diese Änderung jeder Zeit stattfinden.

2.3.4 Das Alert Protokoll

Alerts sind spezifische Arten von Nachrichten. Sie werden durch das SSL Record Layer gesendet. Alerts bestehen aus den zwei Teilen AlertLevel und AlertDescription. Beide Teile sind in einer „single-8-bit-number” kodiert.

SSL 3.0 spezifiziert zwei Arten von Alerten, wie Abb.3 zeigt [3].

Alert level	Level name	Meaning
1	Warning	SSL warnings indicate a problem that is not fatal.
2	Fatal	SSL fatal alerts immediatly terminate the current SSL session.

Abbildung 3: Alert levels

3. WEB SERVICES

Web Services stellen heutzutage einen modernen Ansatz zur Realisierung von verteilten Anwendungen dar. Sie sind ziemlich komplizierte Software-Anwendungen, die mit Hilfe von web-basierten und miteinander zu verknüpfenden Standards entwickelt und aufgebaut werden.

Im Folgenden werden Web Services und Web Standards näher beschrieben.

3.1 Was ist ein Web Service

Ein Web Service ist ein Dienst, der über das Internet verfügbar und mit einem eindeutigen Uniform Ressource Identifier (URI) identifizierbar ist [2]. Web Services benutzen einen standardisierten XML-Nachrichtenaustausch-Mechanismus. Dies ermöglicht bei der Kommunikation eine Unabhängigkeit von bestimmten Betriebssystemen oder Programmiersprache. Veranschaulicht wird diese Tatsache in Abb.4 .

Web Services haben zusätzlich zwei wichtige Eigenschaften: Sie sind selbst-beschreibend und einfach entdeckbar. Mit der ersten Eigenschaft ist gemeint, dass der Web Service eine öffentliche Schnittstelle bereitstellen soll. Diese Schnittstelle könnte beispielsweise eine für humane Benutzer verständliche Dokumentation des Dienstes sein. Ebenfalls muss ein Web Service einfach und schnell gefunden werden, damit interessierte Parteien wie Entwicklern oder andere Web Dienste problemlos auf ihn zugreifen können.

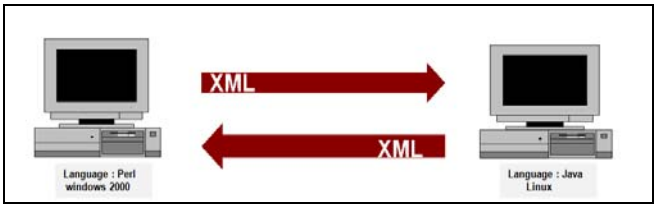

Abbildung 4: Standardisierter XML-Nachrichtenaustausch zwischen zwei verschiedenen Systemen

3.2 Die Architektur von Web Services

Web Services basieren auf einer Service orientierten Architekur (SOA)[2], Dies ist ein sehr verbreiteter Ansatz im Bereich der verteilten Anwendungen, der die Bereitstellung von bestimmten Diensten und Funktionalitäten ermöglicht. Web Services kombinieren daher eine Anzahl von verteilten und objektorientieren Standards um den Autausch von Nachrichten zwischen genau definierten Rollen vorzusehen. Außerdem bietet der sogenannte *Web Service Protocol Stack* (Abb. 6) mit seinen verschiedenen Schichten ein grundlegendes Architektur-modell an, der die einzusetzenden Technologiestandards formal beschreibt.

3.2.1 Rollen und Aktivitäten

Wie in der SOA Spezifikation beschrieben ist, setzt auch die Web Service Architektur die Präsenz von genau drei Rollen mit deren entsprechenden Aktivitäten voraus (siehe Abb.5).

Der Dienstanbieter (Service Provider) ist der Anbieter vom Web Service. Er implementiert seinen Dienst, publiziert ihn auf einem über das Internet erreichbaren Server und dokumentiert ihn anhand von einer öffentlichen und von diversen Applikationen lesbaren Schnittstelle damit die Dienstnutzer leicht auf ihn zugreifen können.

Das Dienstverzeichnis (Service Repository) enthält eine logische Beschreibung zu den veröffentlichen Schnittstellen beispielsweise in Form eines Katalogs. Hier werden die verschiedenen Angaben verwaltet.

Der Dienstnutzer (Service Requestor) ist der Dienstkonsument. Er interagiert mit dem Service Verzeichnis mittels XML-basierten Nachrichten. Im Sinne von Web Service Prinzip sind die Dienstnutzer nichts anderes als reine Softwaresysteme.

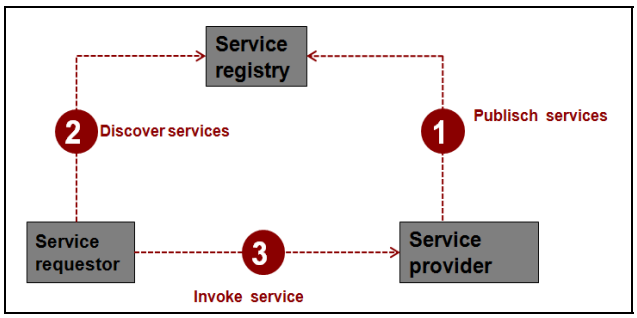

Abbildung 5: Rollen und Aktivitäten in der WS Architektur

3.3 Standards und Technologien

Wie bereits erwähnt bilden die Basistechnologien wichtige Bausteine für die Realisierung von Web Sevices. Diese Standards basiern auf XML und sind im Web Service Protokoll Stack wie folgt beschrieben.

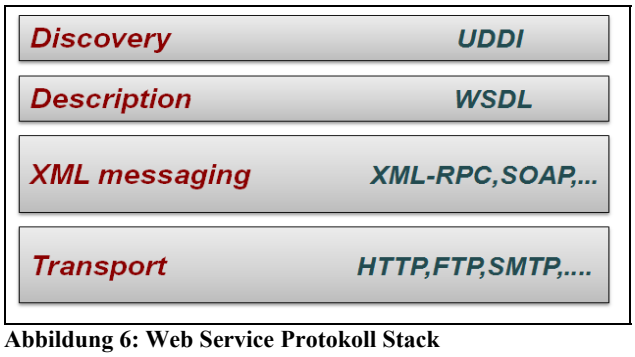

Abbildung 6: Web Service Protokoll Stack

Das Simple Object Access Protocol (SOAP) ist ein Netzwerkprotokoll, das dem Austausch der Daten zwischen zwei verschiedenen Systemen dient. In der Regel wird SOAP für das Remote Procedure Calls durch HTTP verwendet, allerdings sind andere Kommunikationsalternativen möglich (*XML-RPC).* Mittels SOAP kann eine Client-Applikation sehr einfach mit einem Service verbunden werden. Danach kann diese die entsprechend entfernte Methode aufrufen. Es gibt verschiedene andere Technologien (CORBA, DCOM, Java RMI), die die gleiche Funktionalität anbieten [14]. Allerdings besteht die Stärke von SOAP darin, dass die SOAP Nachrichten komplett in XML formuliert sind. Eine totale Plattform- und Programmiersprachen-unabhängigkeit ist daher gewährleistet, welches in Abb. 4 dargestellt ist.

Die Web Services Description Language (WSDL) ist eine Spezifikation zur Beschreibung von Web Services anhand einer allgemeinen XML-Syntax. Mit Hilfe der Metasprache WSDL können also alle wichtigen Informationen zum Aufruf eines Dienstes beschrieben werden. Diese Informationen enthalten die bereitgestellten Funktionen, Daten, Datentypen und Austauschprotokolle eines Web Service. Hauptsächlich wird WSDL in Kombination von SOAP und XML-Schema verwendet.

Die Universal Description, Discovery and Integration (UDDI) [16] ist eine technische Spezifikation zur Beschreibung, Entdeckung und Integration von Web Services. Innerhalb des Web Service Protocol Stack spielt UDDI eine sehr wichtige Rolle. Es erlaubt den Unternehmen, Dienste zum Publizieren und zum Auffinden zu nutzen. Wegen der Verwendung von XML hat ein UDDI Verzeichnis einen Baum als interne Datenstrukur. In dieser Baumstruktur gibt es verschiedene Elemente wie Business-Entities, Busniess-Service, Binding-Template und technische Modelle.

4. Sicherheit für Web Services

Bei der Entstehung des Internets war die Sicherheit keine kritische Frage. In der heutigen Zeit hat das Web viel an Wert gewonnen, besonders im wirtschaftlichen Kontext. Man findet tausende von Geschäftsprozessen einschließlich E-Kommerz, Zahlungsverkehr, Aufträge, Buchungen und viel andere die über das Internet laufen. Allein im Jahr 2008 sind die Umsätze der Online-Händler auf einen Rekordwert von rund 10 Milliarden Euro gestiegen. Daher ist die Gewährleistung der Vertraulichkeit, Verfügbarkeit und Integrität der Dienste erforderlich. Dies erfolgt nur durch die Anwendung von bestimmten Sicherheitmechanismen.

4.1.1 Sicherheit in Web Services

In der SOAP Spezifikation [17] wurden keine expliziten Sicherheitsanforderungen erwähnt. Allerdings gibt es für die Umsetzung von Sicherheit in Web Services eine gute Anzahl von definierten Standards [1]: Darunter findet man den bekannten „Web Service - Security“. Dieser Standard wurde von OASIS entwickelt und liegt zurzeit in der Version 1.2 vor [18]. Es existieren außerdem noch die sogenannte Security Pattern wie Message Inspector, Secure Message Router, Front Door [Steel., 2006]. Wegen der intensiven Nutzung von XML Dokumenten bei der Kommunikation zwischen den beteiligten Systemen kommen auch innovative Methoden zum Einsatz. Dabei setzen sich Technologien wie XML-Encryption bzw. XML-Signaturen durch. XML-Encryption Spezifikation beschreibt eine Reihe von Möglichkeiten zur Ver- und Entschlüsselung von XML-Dokumenten, dagegen definiert die XML-Signatur Spezifikation eine XML-Schreibweise für digitale Signaturen [18].

4.1.2 Einsatz von SSL/TLS in Web Services

Ein SOAP basierter Web Service bietet seine Funktionalität über das Web via XML-Nachrichten an, die mittels SOAP übermittelt werden. Eine häufige Kombination ist SOAP über HTTP und TCP, daher ist eine Verwendung von SSL/TLS möglich [4]. (Abb.7)

SSL/TLS bietet auf der Transportebene Vertraulichkeit und Authentifizierung an. Für den Nachrichtenaustausch zwischen genau zwei Partnern stellt SSL/TLS einen guten Lösungsansatz wegen der schnellen Umlauf und Struktur der Nachrichten dar. Ein bekanntes Beispiel dafür ist die Verschlüsselung der Nachrichten bei Banken-Transaktionen.

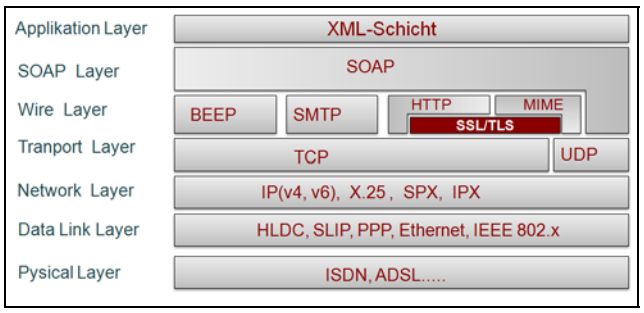

Abbildung 7: Einsatz von SSL/TLS in Web Services

Allerdings sehen die Experten, dass dieses Protokoll für eine umfassende Sicherung von Web Services nicht ausreichend ist. Es treten daher folgende Probleme bei dem Einsatz von SSL/TLS zur Sicherung der Web Dienste auf:

Kosten: Die Verwaltung der Umgebung und die Anschaffung der Technologie für jeden Service Konsumenten und jede Web-Service Kombination ist schwer und kostspielig.

Leistung (Performance): Das Anschaffen und der Aufbau von SSL Verbindungen für jede einzelne Nachricht kann Performanz-Problemen verursachen.
Eine große Anzahl von Teilnehmern bedeutet auch einen großen Aufwand bei der Schlüsselerstellung und Verwaltung.

Sicherheit: Es besteht ein Sicherheitsrisiko bei der Übermittlung von ungesicherten Web Services. Es ist auch keine Certificate Revokation List (CRL) für jeden Nutzer vorhanden, mit der die Ungültigkeit von verschiedenen Zertifikate festgestellt werden können. Die Vertraulichkeit der Daten ist lediglich zwischen zwei Knoten und nicht etwa *von Beginn bis Ende der Übermittlungskette* gewährleistet. Die folgende Abbildung illustriert diesen Kontext:

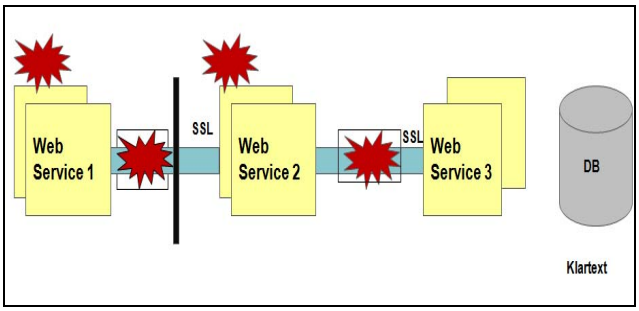

Abbildung 8: Web Service Übermittlungskette

5. Zusammenfassung

Diese Arbeit ist eine grundlegende Einführung in die Kryptographische Verfahren SSL/TLS und ihre Einsatz in Web Services. SSL/TLS stellt eine der verbreitesten Technologien zur Sicherung der Komunikationskanälen. Dies ist auf die Effizienz und die gute Strukturierung der enthaltenen kryptographischen Komponenten zurückzuführen.

Der Einsatz von Web Services ist ein moderner Trend in der Internet Welt. Allerdings treten diese Web Dienste häufiger in isolierten Umgebungen wie das Intranet. Der Grund dafür liegt an der immer noch umstrittenen Rolle von Sicherheitsverfahren (z.B SSL) zur Erfüllung der strikten Sicherheitsanforderungen .

6. Literatur

[1] M.Bichler-TUM Vorlesungsskript: Internetbasierte Gescäftssysteme: http://ibis.in.tum.de/teaching/ws08_09/index.htm

[2] E. Cerami- Web Services Essentials

[3] S. Garfinkel, G. Spafford – Web Security, Privacy & Commerce

[4] P.Kumar –The pros and cons of securing Web Services with SSL

[5] K.Manhart- Sicherheit bei Web Services http://www.tecchannel.de/webtechnik/soa/479383/sicherheit_bei_web_services/

[6] E. Rescorla - SSL and TLS Design and Building Secure Systems: *HTTP over SSL*, Addison-Wesley,2001 ISBN 0-201-61598-3 pp.291-307

[7] J.Schlichter-TUM Vorlesungsskript: Verteilte Anwendungen

[8] SSL Concepts: Hitsory of SSL http://publib.boulder.ibm.com/iseries/v5r2/ic2924/index.htm?info/rzain/rzainhistory.htm

[9] Josh Benaloh, Butler Lampson, Daniel Simon, Terence Spies, Bennet Yee, Microsoft Corp. The Private Communication Technology(PCT) Protocol. Internet Draft

[10] Nau Okamoto, Shigetomo Kimura , Yoshihiko Ebihara,*"An Introduction of Compression Algorithms into SSL/TLS and Proposal of Compression Algorithms Specialized for Application"*, Advanced Information Networking and Applications, 2003 , 17th International Conference.

[11] Transport Secure Layer. Wikipedia. Available at: http://de.wikipedia.org/wiki/Transport_Layer_Security

[12] Repges Markus , *"Einfüfrung in SSL"* Available at: http://www.repges.net/SSL/ssl.html

[13] Diffie, W. and Hellman, M.E.(1976) *"New Directions in Cryptograpy"*, IEEE Transactions on Information Theory (22:6), pp.644-54

[14] Remote Procedure Call. Wikipedia. Available at : http://de.wikipedia.org/wiki/Remote_Procedure_Call

[15] Web Services Description Language (WSDL) Version 2.0 Part1: Core Language http://www.w3.org/TR/wsdl20/

[16] UDDI Version 2 Specifications, OASIS- Committee Specifications http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.pdf

[17] SOAP Version 1.2, W3C Recommendation (Second Edition) 27 April 2007: http://www.w3.org/TR/soap/

[18] OASIS Web Services Security (WSS) TC Available at: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

[19] XML Signature Syntax and Processing (Second Edition) W3C Recommendation 10 June 2008. Available at : http://www.w3.org/TR/xmldsig-core/