

Verschlüsselung im WWW

Die sichere Übertragung von Daten im Internet gewinnt immer mehr an Bedeutung. Nicht nur für Anwendungen wie Online-Shopping oder Banküberweisungen sind sichere Kommunikationswege absolut unerlässlich. Auch das Anbieten von vertraulichen Informationen, die nur für einen bestimmten Personenkreis bestimmt sind, oder die Publikation elektronischer Dokumente erfordern den Einsatz von Sicherheitsmaßnahmen, um die Daten vor unbefugtem Zugriff oder unerlaubter Veränderung zu schützen. In diesem Artikel sollen Möglichkeiten für sichere Verbindungen im WWW aufgezeigt werden. Nach einer Einführung zum Sicherheitsprotokoll SSL und den dabei verwendeten Zertifikaten soll anhand eines praktischen Beispiels die Implementation eines sicheren Webservers demonstriert werden.

Sichere Kommunikation

Schon im ersten Absatz wurde von sicheren Verbindungen gesprochen. Welches sind nun konkrete Anforderungen an solche Verbindungen? Zuallererst ist die *Vertraulichkeit* des Datenaustausches zu nennen. Niemand außer den Kommunikationspartnern soll in der Lage sein, die übertragenen Informationen „mitzuhören“, z. B. durch Aufzeichnen des Datenstroms auf dem Netzwerk. Mit den derzeit verwendeten Internetprotokollen werden ohne Zusatzmaßnahmen selbst solche sensible Daten wie Paßwörter unverschlüsselt über das Netz übertragen, so daß sie schon mit mittelmäßigem technischen Geschick mitgeschnitten werden können. Eine zweite Forderung ist die Sicherung der *Integrität* der übermittelten Daten, d. h. sie sollen so beim Empfänger ankommen, wie sie abgeschickt wurden. So hat das simple Anhängen von sechs Nullen an den Betrag einer Banküberweisung sicherlich fatale Folgen für den ahnungslosen Absender. Zuletzt ist noch die Forderung nach der *Authentizität* von Verbindungen im Internet zu stellen. Sowohl der Empfänger als auch der Absender wollen sicher sein, wirklich mit dem gewünschten Partner und nicht mit einem Dritten zu kommunizieren. Wer sich z. B. die aktuellen Aktienkurse von einem WWW-Server lädt, möchte natürlich sichergehen, daß er wirklich mit dem Server der Börse verbunden ist, um sich auf die Zahlen verlassen zu können. Neben diesen technischen Anforderungen sollte allerdings nicht vergessen werden, daß ausgefeilte Protokolle oder aufwendige Software gar nichts nutzen, wenn sie so kompliziert sind, daß die Anwender weder die Hintergründe (wenigstens in Grundzügen) durchschauen noch die entsprechenden Programme verstehen können. *Einfache Bedienbarkeit der Programme* und *anwenderorientierte Vermittlung der Sicherheitsprobleme im Internet* sind Grundvorausset-

zungen für die breite Anwendung von sicherer Kommunikationssoftware.

Die im letzten Absatz genannten technischen Forderungen lassen sich mit unterschiedlichen kryptographischen Verfahren erfüllen. So werden vertrauliche Kommunikationswege mit Hilfe von symmetrischen Verfahren verschlüsselt. Der dafür notwendige Schlüssel wird mit Hilfe asymmetrischer Verfahren ausgehandelt. Digitale Unterschriften sichern die Authentizität der Kommunikationspartner. Zur Sicherung der Integrität der Daten werden Message-Digest-Verfahren verwendet. Die Erläuterung der theoretischen Grundlagen dieser Verfahren würde den Rahmen des Artikels sprengen, jedoch sollte der Leser für das weitere Verständnis mit den wesentlichen Eigenschaften vertraut sein. Eine kurze Einführung findet man unter [1]. Detaillierte Informationen werden auf dem Server [2] angeboten.

Zertifikate

Grundlegend für das Zustandekommen authentischer Verbindungen sind Zertifikate. Diese können als „digitale Personalausweise“ angesehen werden, da sie es ermöglichen, auf elektronischem Wege gegenseitig die Identität der Kommunikationspartner zu überprüfen. Sie lösen das Problem, bei Anwendung der asymmetrischen Verschlüsselung die Zugehörigkeit eines public key zu einer Person sicherzustellen. Zertifikate enthalten allgemeine Informationen zur Person oder Einrichtung (z. B. Name, Adresse, E-Mail-Adresse usw.) und den eigenen public key. Diese Informationen werden mit einer Prüfsumme versehen und mit dem private key einer *trusted third party* (auch certificate authority oder kurz CA) unterschrieben, also einer Institution, der beide Kommunikationspartner vertrauen. Das bedeutet, der public key der CA ist entweder von einer übergeordneten CA unterschrieben oder weithin publiziert und bekannt, so daß Betrug ausgeschlossen werden kann. Weiterhin muß die CA so sorgfältig arbeiten, daß sie erstens ihren eigenen private key äußerst sorgfältig schützt und zweitens die Identität von Nutzern, die ein Zertifikat beantragen, genau prüft. Zertifikate werden nach dem X.509-Standard kodiert. Informationen zum Nutzer selbst werden als *Distinguished Name* (DN) bezeichnet. Hierbei werden in genau beschriebenen Feldern z. B. der Wohnort, der Staat oder der eigentliche Name eingetragen. Jedes dieser Felder hat eine Kurzbezeichnung. So könnte der Autor folgenden DN besitzen:
/CN=Daniel Ohst/O=Humboldt-University/
OU=Computer Center/L=Berlin/C=Germany.

SSL-Protokoll

Insbesondere für die Anwendung im World Wide Web hat Netscape das Secure Socket Layer (SSL)-Protokoll entworfen. Es setzt auf dem TCP/IP-Protokoll auf und kann deshalb auch für andere Internetdienste, wie z. B. FTP oder telnet, verwendet werden. Das Protokoll erlaubt verschlüsselte Verbindungen, Authentisierung mit Zertifikaten nach X.509 von Server und Client sowie die Sicherung der Integrität der Daten mit Message-Digest-Verfahren. Dabei werden zahlreiche kryptographische Verfahren unterstützt. SSL initialisiert eine sichere Verbindung mit einem Handshake-Protokoll zwischen den Kommunikationspartnern, das zulässige Algorithmen aushandelt, Zertifikate austauscht und zuletzt einen session key für die Verschlüsselung festlegt. Ausführliche Informationen hierzu erhält man unter [3].

Auf der Clientseite wurde SSL in die Browser Netscape Navigator und Internet Explorer (ab Version 3) integriert. Beide unterstützen verschlüsselte Verbindungen mit einem SSL-basierten Webserver und die Verwaltung von Zertifikaten. In den weiteren Ausführungen wird nur noch auf die konkrete Implementation der aktuellen Version 4.03 des Navigators Bezug genommen, da der Funktionsumfang und die Nutzung der SSL-Dienste zwischen den verschiedenen Browsern und Versionen zum Teil stark differieren. In [4] und [5] findet man weitergehende Hinweise zu diesen Unterschieden, auf die aus Platzgründen hier keine Rücksicht genommen werden kann.

Nach der Installation von Netscape sind standardmäßig unter dem Punkt *Security Info/Certificates* Zertifikate der bekannten internationalen CAs wie z. B. Verisign integriert. Mit einem WWW-Server, der ein von einer dieser CAs unterschriebenes Zertifikat besitzt, kann also sofort eine sichere Verbindung aufgebaut werden. Diese CA-Zertifikate werden im Navigator in die Gruppe *Signers* eingeordnet. Unter *Websites* können Nutzer selbstunterschriebene Zertifikate eines WWW-Servers akzeptieren. Unter *People* und *Yours* können sogenannte client certificates geladen werden, die an Personen und nicht an Server gebunden sind. Ab der Version 4 des Navigators werden Zertifikate auch für das Versenden sicherer E-Mail nach dem S/MIME-Standard verwendet.

Unter Security Info/Navigator können noch einige Einstellungen für die Aushandlung sicherer Verbindungen (z. B. verwendete Algorithmen) vorgenommen werden. Hierbei gelten immer noch US-Exportbeschränkungen, so daß größere Schlüssellängen von z. B. 128 bit nicht genutzt werden können, obwohl die Algorithmen implementiert sind.

SSL-Implementationen

Wer selbst einen WWW-Server aufbauen will, zu dem sichere Verbindungen möglich sind, hat die Wahl zwischen der Implementation auf der Basis frei verfügbarer Produkte oder einer kommerziellen Lösung. Beide haben Vor- und Nachteile.

Zahlreiche Firmen, darunter selbstverständlich auch die beiden großen Browserhersteller, bieten entsprechende Server an. Ein bekannter Vertreter ist der Enterprise-Server von Netscape. Eine Sonderrolle nimmt der Secure Server von Stronghold ein, der kommerziell vertrieben wird, jedoch auf freier Software basiert. Die Kosten für einen kommerziellen Server belaufen sich auf ca. \$ 500 - 2000. Weiterhin schlägt ein Serverzertifikat einer anerkannten CA mit ca. \$ 400 pro Jahr zu Buche. Wer Clientzertifikate zur Autorisierung benutzen will, muß noch einmal ca. \$ 20 pro Nutzer und Jahr dazurechnen.

Das frei verfügbare SSLeay-Paket von Eric Young und Tim Hudson hat sich als Quasistandard bei der Nutzung von SSL durchgesetzt. Es ist eine umfangreiche Sammlung kryptographischer Algorithmen und enthält Programme und Skripte zur Verwaltung eigener CAs und Zertifikate. SSLeay ist im Sourcecode für UNIX- und Windowssysteme zu erhalten. Um geeignete Applikationen nutzen zu können, wurden Patches für die wichtigsten Internetdienste erarbeitet, so für den hervorragenden Apache-Webserver, aber auch für telnet- oder ftp-Dienste.

Einer Firma, die ihren Katalog nun auch im Internet anbieten und den Kunden Gelegenheit zur sicheren Onlinebestellung geben will, ist sicherlich mit einem kommerziellen Angebot am besten gedient. Die Server besitzen grafische Oberflächen zur Administration, und das Einbinden des Serverzertifikats ist auch für Laien durchführbar. Allerdings ist über einen Zeitraum von drei Jahren mit Kosten für Software, Updates und Zertifikate von ca. \$ 4000 zu rechnen, was schon ein recht beachtlicher Betrag ist.

Wer für seine Einrichtung sichere Weblösungen installieren will, wenig oder gar kein Geld, aber dafür Kenntnisse in der UNIX-Administration und im Sicherheitsmanagement besitzt (z. B. nach dem Lesen dieses Artikels), kann auf freie Softwarelösungen zurückgreifen. Eine solche soll im nächsten Abschnitt praktisch demonstriert werden. Hierbei ist jedoch ein erhöhter Personalbedarf einzukalkulieren, da das Betreiben einer eigenen CA klar definierte Handlungsabläufe und Sicherheitsmaßnahmen erfordert.

Ein sicherer WWW-Server

Auf der Basis frei verfügbarer Software soll im folgenden ein Beispielprojekt beschrieben werden, das zum Ziel hat, verschlüsselte Verbindungen zu einem Webserver aufzubauen. Weiterhin sollen sich Server

und Clients gegenseitig authentifizieren können. Voraussetzung für die Implementation ist die Verwaltung einer eigenen CA, die Certificate Signing Requests entgegennimmt und Serverzertifikate ausstellt. Nutzer sollen über ein Formular Clientzertifikate beantragen können. Nach einem bestimmten Verfahren wird dann die Identität geprüft, der Request durch die CA unterschrieben und an den Nutzer zurückgesandt.

Als Hardwareplattform wird ein PC mit einer aktuellen Linuxdistribution (Kernel 2.0.30) vorausgesetzt. Alle Installationsanweisungen sind jedoch ohne Probleme auf andere Plattformen und Betriebssysteme, z. B. SUN Solaris 2.5.1, zu übertragen. Es kommt das SSLeay-Paket in der Version 0.8.1 zur Anwendung, was nach der Kompilierung unter `/usr/local/ssl` installiert sein sollte. Als Webserver kommt Apache in der Version 1.2.4 zum Einsatz. Er muß vor dem Übersetzen mit der Version 1.11 von Apache-SSL gepatcht und unter `/usr/local/httpd` installiert werden. Als Client wurde der Netscape Navigator 4.03 verwendet. Wenn man einige kleine Bugs in Kauf nimmt, kann auch die Version 3.01 eingesetzt werden. Der Internet Explorer unterstützt zwar auch alle SSL-Features, unterscheidet sich jedoch zum Teil stark in der Zertifikatsverwaltung und soll deshalb in diesem Beispiel keine Verwendung finden. Auch kann nicht auf jedes Detail der Konfiguration (insbesondere des WWW-Servers) eingegangen werden. Die obengenannte Software und umfangreiche Installationshinweise kann man sich unter [4] und [6] herunterladen.

CA-Verwaltung

Der Rechner, auf dem sich die CA-Schlüssel befinden und auf dem Zertifikate ausgestellt werden, ist idealerweise physikalisch abgeschirmt und besitzt keinen Netzanschluß. Es ist klar, daß ein Eindringling, der in der Lage ist, den CA-Rechner und die dazugehörigen Daten zu kompromittieren, die Sicherheit des ganzen Systems in Frage stellt. Datentransfers sollten deshalb per Diskette stattfinden. Zu Testzwecken ist jedoch die Installation auf dem Rechner des WWW-Servers ratsam, um Fehlkonfigurationen schneller beheben zu können.

Der erste Schritt nach der Installation von SSLeay sollte das Editieren der Datei `ssleay.cnf` im lib-Verzeichnis (immer bezogen auf das Installationsverzeichnis `/usr/local/ssl`) sein. Hier werden Voreinstellungen zu Schlüssellänge, Gültigkeitsdauer und Bestandteilen des DN vorgenommen. Der Eintrag `dir` sollte in einen absoluten Pfad geändert werden. Policies legen z. B. fest, ob Zertifikate nur für Nutzer der eigenen Abteilung oder auch der gesamten Einrichtung ausgegeben werden können. Weitere Informationen hierzu erhält man in [5].

Mit dem Aufruf von `bin/CA.sh - newca` erzeugt man eine Verzeichnisstruktur mit dem gewählten Namen und

ein Schlüsselpaar. Die Passphrase sollte möglichst lang sein und ist der einzige Schutz des private key. Sie ist deshalb unter allen Umständen geheimzuhalten und niemals auf dem Rechner selbst zu speichern. Danach wird der DN der CA festgelegt. Der Common Name könnte z. B. der Domainname Ihrer Einrichtung sein (`rz.hu-berlin.de`). Damit der Browser alle Zertifikate, die von dieser CA ausgestellt wurden, ohne weitere Sicherheitsabfragen akzeptiert, muß das Zertifikat im Browser installiert werden. Dazu muß in `srm.conf` von Apache die Zeile `AddType application/x-x509-ca-cert .ca` eingetragen werden. Das CA-Zertifikat, das als `cacert.pem` im ASCII-Format vorliegt, muß zuerst mit der Kommandozeile `bin/x509 -in cacert.pem -out cacert.der -outform DER` in ein für den Navigator verständliches Binärformat konvertiert werden. Dieses kann dann unterhalb des Dokumentenverzeichnisses des Servers abgelegt und heruntergeladen werden, was auch unverschlüsseltem Wege geschehen kann. Man sollte den Nutzern jedoch auf jeden Fall auf einem sicheren Wege (z. B. in einem Rundschreiben oder einer Zeitung) die Prüfsumme des Zertifikats zur Kontrolle übermitteln. Nach einem kurzen Dialog und der Vergabe eines Namens für die CA erscheint dieser auch schon in der Liste der *Signers* (Abb. 1).

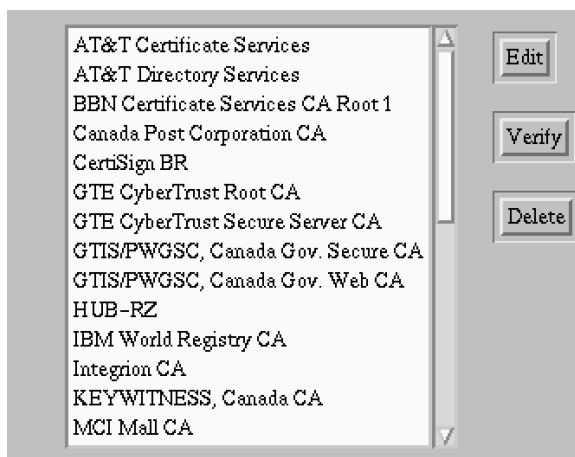


Abb.1: Akzeptierte CA-Zertifikate in Netscape 4.03

Ein selbsterstelltes CA-Zertifikat im Navigator könnte dann z. B. wie in Abb. 2 aussehen.

Der nächste Schritt ist die Erstellung eines Serverzertifikats. Mit dem Befehl `bin/CA.sh -newreq` wird ein Certificate Signing Request (CSR) und ein neues Schlüsselpaar erzeugt. In der Datei `newreq.pem` befindet sich dann der CSR und der verschlüsselte private key. Mit `bin/CA.sh -sign` wird das eigentliche Zertifikat in der Datei `newcert.pem` erzeugt, nachdem das Paßwort für den CA private key und die Angaben für den DN geprüft wurden. Dieses Zertifikat kann nun im WWW-Server verwendet werden.

Bei der Apache-Konfiguration ist es günstig, den sicheren Server auf Port 443 als VirtualHost zu dekla-



Abb. 2: CA-Zertifikat im Netscape Navigator

rieren. Die SSL-Optionen, wie Pfade zu CA- und Serverzertifikaten, sind auszufüllen. Bei jedem Start des Servers ist nun das Paßwort für den private key des Zertifikats einzugeben. Zum jetzigen Zeitpunkt ist es schon möglich, sichere HTTP-Verbindungen aufzubauen, bei denen sich der Server gegenüber dem Nutzer durch Angabe seines Zertifikats authentisiert. Auch Anwender, die das CA-Zertifikat nicht geladen haben, können das Zertifikat nach einem kurzen Netscape-Dialog in die Gruppe *Websites* aufnehmen und ebenfalls sicher kommunizieren.

Client-Zertifikate

Zuletzt soll noch das herkömmliche Authentisierungsschema für Webseiten (Name/Paßwort) durch die Anwendung von Clientzertifikaten verbessert werden. Hierzu ist etwas mehr Arbeit nötig, da ein Schlüsselpaar im Navigator erzeugt, daraus ein CSR generiert und an die CA-Verwaltung geschickt werden muß. Diese prüft die Identität, stellt das Zertifikat aus und schickt Informationen zu dessen Download an den Anwender zurück.

Zur Beantragung eines Zertifikats muß der Nutzer ein HTML-Formular ausfüllen, in das er seine persönlichen Angaben einträgt. Ein Beispiel findet man unter [7] als cert.html. Geändert werden muß der URL des Scripts im FORM-Tag. Nach dem Abschicken erzeugt der Navigator durch das spezielle <KEYGEN>-Tag ein Schlüsselpaar. Auch hier steht durch Exportbeschränkungen nur die Schlüssellänge 512 bit zur Verfügung. Der private key wird in einer Datenbank gespeichert und sollte durch ein Paßwort geschützt werden. Danach wird ein CSR erzeugt und an das angegebene Perl-Skript weitergeleitet. Auch hier kann man als Anregung für eigene Implementationen das Programm *generate.pl* unter [7] nutzen, das bezüglich der Adressen und Pfade an die eigenen Gegebenheiten angepaßt werden muß. Es wird eine Datei im /tmp-Verzeichnis

erzeugt, die den CSR in einem speziellen Netscape-Format enthält. Weiterhin wird im Verzeichnis *\$HOME/requests* des Nutzers *webadm* eine Datei angelegt, die den CSR sowie die generierte Zufallszahl und die Telefonnummer enthält. Die Identitätsprüfung könnte nun darin bestehen, daß der Nutzer zurückgerufen wird und die richtige Zufallszahl nennen muß. Danach wird das Zertifikat erzeugt und dem Nutzer der URL mitgeteilt. Eine einfachere Lösung besteht darin, den gesamten Prozeß zu automatisieren und eine E-Mail mit den Downloadinformationen zu versenden, was natürlich keiner echten Identitätsprüfung entspricht. Eine aufwendige, aber sehr sichere Variante ist das Einreichen von notariell beglaubigten Identitätsunterlagen. In der Praxis sieht es meist so aus, daß für Clientzertifikate unterschiedliche Sicherheitslevel angeboten werden und jeder selbst entscheiden kann, welche Sicherheitsanforderungen bezüglich der Zertifikate seine Anwendung benötigt.

Aus der von *generate.pl* erzeugten Datei kann mit dem Kommando *ca -spkac „CSR-Datei“ -out „Clientcert.cln“* das Zertifikat für den Nutzer erstellt werden, das dann unterhalb der Dokumentenhierarchie des WWW-Servers erreichbar sein muß. Zum Download muß noch der MIME-Type *application/x-x509-user-cert.cln* in die Datei *srm.conf* eingetragen werden. Nach erfolgreichem Laden steht das neue Zertifikat unter der Gruppe *Yours* im Netscape Navigator.

Um nun auch Client-Zertifikate für die Autorisierung für Seiten oder Verzeichnisse auf dem WWW-Server nutzen zu können, müssen noch einige SSL-Optionen in der Datei *httpd.conf* gesetzt werden. Der Pfad zu dem oder den CA-Zertifikaten muß entsprechend eingetragen und die Optionen *SSLFakeBasicAuth* und *SSLVerifyClient* gesetzt sein. Eintragungen in *access.conf* oder *.htaccess* werden entsprechend der herkömmlichen Autorisierung vorgenommen. Unter-

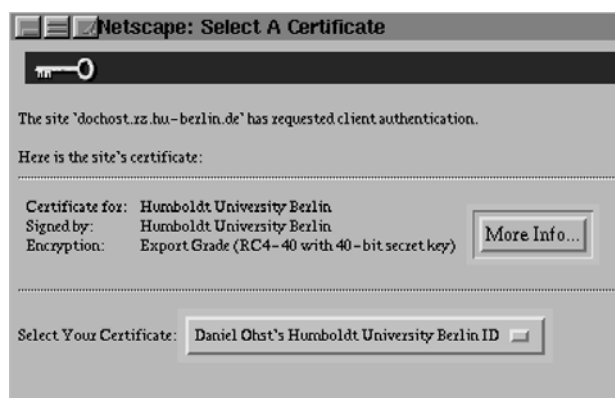


Abb. 3: Anforderung eines Clientzertifikates

schiedlich ist der Aufbau der Accountdateien. Anstatt des Usernamens wird der entsprechende DN eingetragen. Ein Paßwort ist nicht mehr notwendig, so daß dort der Standardstring *xxj3IZMTZzkVA* verwendet wird,

um den syntaktischen Anforderungen zu genügen. Wenn nun der Navigator auf eine Seite zugreifen will, die die Angabe eines Clientzertifikats erfordert, erscheint eine ähnliche Aufforderung wie sie in Abb. 3 dargestellt ist.

Zum Abschluß dürfen natürlich auch einige kritische Anmerkungen zur verwendeten Lösung nicht fehlen. Zum einen sind die durch die Exportbeschränkungen zugelassenen Schlüssellängen (symmetrisch 40 bit) für hohe Sicherheitsanforderungen keinesfalls mehr als ausreichend anzusehen. Hier könnte die Verwendung von SSL-Proxies helfen [8]. Zum anderen ist der Aufbau einer CA aufwendig, insbesondere in bezug auf das strikte Einhalten der Sicherheitsvorkehrungen. Möglich wäre der Anschluß an eine übergeordnete CA, z. B. die DFN-PCA [9]. Leider werden in diesem Projekt z. Z. nur PGP-Schlüssel unterschrieben. Zertifikate nach X.509 für SSL-Anwendungen sind erst in Vorbe-

reitung. Aus diesem Grunde wird das RZ noch in diesem Jahr den Testbetrieb einer eigenen CA für die Universität aufnehmen.

Weiterführende Informationen:

- [1] <http://www2.rz.hu-berlin.de/~h0444saa/rdi/>
- [2] <http://www.cs.hut.fi/crypto/>
- [3] <http://home.netscape.com/eng/ssl3/>
- [4] <http://www.psy.uq.oz.au/~ftp/Crypto/>
- [5] <http://www.opengroup.org/RI/www/prism/wwwj/>
- [6] <http://www.apache.de/>
- [7] <http://www2.rz.hu-berlin.de/~h0444saa/ssl/>
- [8] <http://stronghold.ukweb.com/>
- [9] <http://www.cert.dfn.de/dfnpca/>

Daniel Ohst