



Kryptographie in Notes/Domino – Pragmatische Einführung für Administratoren

11. September 2007

Thomas Bahn
assono GmbH
tbahn@assono.de
<http://www.assono.de>
+49/4307/900-401

© 2007 – assono GmbH



Agenda

- Organisatorisches

- Pragmatische Einführung in Kryptographie
 - Symmetrische Verschlüsselung
 - Asymmetrische Verschlüsselung
 - Telefonbuch für öffentliche Schlüssel
 - „Trick 17“
 - Signaturen



Agenda

- Zertifikate und ID-Dateien
 - Zertifikate
 - Zertifizierer-ID
 - Registrierung neuer Benutzer & Server
 - Notes-Gegenzertifikate
 - ID-Dateien
 - Passwort-Wiederherstellung
 - Authentifizierung am Server



Agenda

- Verschlüsselung und Signaturen
 - Verschlüsselung
 - Netzwerk
 - Datenbanken
 - E-Mails (Notes-intern)
 - Felder in Dokumenten
 - Signaturen
 - E-Mails
 - Dokumente
 - Abschnitte



Agenda

- Internet
 - SSL + HTTP = HTTPS
 - SSL-Client-Zertifikate
 - Sichere E-Mails im Internet: S/MIME
 - Internet-Gegenzertifikate

- Sonstiges



Organisatorisches

Pragmatische Einführung in Kryptographie

Zertifikate und ID-Dateien

Verschlüsselung und Signaturen

Internet

Sonstiges



Organisatorisches

- Wer bin ich?
- Thomas Bahn, IT-Berater, Dipl.-Math., 36 Jahre, verheiratet
- Mitgründer und -inhaber der assono GmbH
- seit 1997 entwickle ich mit Java und RDBMS, z. B. Oracle (OCP)
- seit 1999 mit IBM Lotus Notes/Domino (IBM Certified Advanced Application Developer – Lotus Notes/Domino R4 - 7)
- Mein Schwerpunkt liegt auf Anwendungen mit Schnittstellen zu anderen Systemen, z. B. RDBMS, SAP R/3, und interaktiven Web-Anwendungen
- auch Administrator (IBM Certified System Administrator – Lotus Notes/Domino 6 - 7)



Organisatorisches

- bitte zum Schluss Bewertungsbögen ausfüllen
- Zwischenfragen erwünscht!
- bitte Handys aus- oder stummschalten



Organisatorisches

Pragmatische Einführung in Kryptographie

Zertifikate und ID-Dateien

Verschlüsselung und Signaturen

Internet, Protokolle & Server-Zugriff

Sonstiges



Symmetrische Verschlüsselung

- „symmetrisch“ weil mit dem **gleichen** Schlüssel ver- und entschlüsselt wird
- Algorithmen performant
- „Brute-Force-Angriffe“ (alle Schlüssel ausprobieren) auch!
- Schlüssellänge bestimmt Sicherheit: länger ist besser ;-)
- Problem: Wie verteilt man den Schlüssel?
Man braucht einen „sicheren Kanal“



Asymmetrische Verschlüsselung

- „asymmetrisch“, weil **zwei verschiedene** Schlüssel zum Ver- und Entschlüsseln verwendet werden
- relativ langsam (im Vergleich zu symmetrischen Verfahren)
- kürzere Schlüssel für vergleichbare Sicherheit
- zwei Schlüssel:
 - privater Schlüssel: ist geheim, darf nur der Besitzer kennen
 - öffentlicher Schlüssel: darf und soll (!) öffentlich bekannt sein
- Verschlüsselung mit dem öffentlichen Schlüssel: kann also jeder
- Entschlüsselung mit dem privaten Schlüssel: kann nur sein Besitzer



Asymmetrische Verschlüsselung (forts.)

- Falltür-Funktion:
 - in die eine Richtung geht es einfach...
 - aber man kann mit dem verschlüsselten Text (Chiffre) und dem öffentlichen Schlüssel nicht auf die ursprüngliche Information zurück kommen.
- Asymmetrische Verfahren sind auch ein bisschen symmetrisch: Man kann auch mit dem privaten Schlüssel verschlüsseln. Das Ergebnis kann dann mit dem öffentlichen Schlüssel wieder entschlüsselt werden.



Telefonbuch für öffentliche Schlüssel

- öffentlicher Schlüssel wird in Verzeichnissen bekannt gemacht
- gebraucht wird eine Art „Telefonbuch“ mit Name → Schlüssel-Einträgen
- Domino-Verzeichnis ist so ein Verzeichnis:

Person: **Thomas Bahn/assono** tbahn@assono.de

Basics | Work/Home | Other | Miscellaneous | **Certificates** | Roaming | Administration

Notes Certificates | Internet Certificates | Flat Name Key

Notes Certificate(s)

Notes certificate:	Present
Notes certified public key:	03002E02 A040F2C8 08601617 00020654 C5C27C03 60030200 01208600 63CF2000 156F2560 024F6002 63CF2000 146F2500 6C732000 597D25C1 01A07700 63CF2000 156F2560 024F6002 63CF2000 146F2500 6C732000 597D25C1 4F3D6173 736F6E6F



„Trick 17“

- Symmetrische Verschlüsselung ist schnell, hat aber das Problem der Schlüsselverteilung.
- Asymmetrische Verschlüsselung hat kein Problem bei den Schlüsseln, ist aber deutlich langsamer (bei gleicher Sicherheit)
- Ein häufiger Trick besteht darin, beides zu kombinieren:
 - Es wird zufällig ein langer Schlüssel für die symmetrische Verschlüsselung vom Sender erzeugt.
 - Dieser wird asymmetrisch verschlüsselt mit dem öffentlichen Schlüssel des Empfängers und an diesen gesendet.
 - Er (und nur er) kann ihn entschlüsseln. So teilen beide Seiten den gleichen geheimen Schlüssel.
 - Die restliche Kommunikation wird symmetrisch mit diesem Schlüssel verschlüsselt.



Signaturen

- Verschlüsseln: nur bestimmte Personen dürfen Information sehen
- Signaturen: Beweis, dass Information
 1. wirklich von bestimmter Person stammt und
 2. nicht verändert wurde
- Hash-Funktion erzeugt Prüfsumme:
 - macht aus einem langen Text eine Zahl fester Länge
 - ergibt eine ganz andere Zahl, wenn der Text nur ganz wenig verändert wird
- Signieren: Hash-Wert des Textes mit privaten Schlüssel verschlüsseln (kann also nur der Besitzer des privaten Schlüssels)
- Prüfen: verschlüsselten Hash-Wert mit öffentlichen Schlüssel entschlüsseln und mit selbst berechneten Hash-Wert vergleichen



Signaturen (forts.)

- Prüfen: verschlüsselten Hash-Wert mit öffentlichen Schlüssel entschlüsseln und mit selbst berechneten Hash-Wert vergleichen
- Nur der Besitzer des privaten Schlüssels kann die Prüfsumme so verschlüsseln, dass er mit dem öffentlichen Schlüssel entschlüsselt werden kann.
- Wurde der Text nach der Signatur verändert, verändert sich auch sein Hash-Wert, so dass beim Vergleich die Änderung entdeckt wird.



Organisatorisches

Pragmatische Einführung in Kryptographie

Zertifikate und ID-Dateien

Verschlüsselung und Signaturen

Internet, Protokolle & Server-Zugriff

Sonstiges



Zertifikate

- „Ein Zertifikat ist ein elektronischer Stempel zur Identifizierung eines Benutzers oder Servers.“ [Ebel2004], S. 322
- technischer: Ein Zertifikat ist im Wesentlichen die Signatur von Benutzerinformationen.
- Zertifikate...
 - werden von zentraler Stelle, der Certification Authority (CA = Zertifizierungsstelle) ausgestellt und können mit ihrem öffentlichen Schlüssel geprüft werden.
 - sind normalerweise zeitlich nur beschränkt gültig.
 - beweisen, dass die Benutzerinformationen „echt“ und unverändert sind.



Zertifizierer-ID

- Bei Notes/Domino: Bei der Installation des ersten Servers einer Notes-Domäne wird eine zentrale Zertifizierungsstelle, die Zertifizierer-ID erstellt.
- Zertifizierer-ID enthält privaten und öffentlichen Schlüssel. Ihr öffentlicher Schlüssel steht auch im Domino-Verzeichnis.
- OU-Zertifizierer funktionieren analog.



Registrierung neuer Benutzer & Server

- Beim Erstellen wird für alle Benutzer und Server beim Erstellen ein Zertifikat von der benutzten (OU-)Zertifizierer-ID ausgestellt und in der ID-Datei und dem Domino-Verzeichnis (Personen-/Server-Dokument) gespeichert.
- Sie können damit beweisen, dass sie wirklich mit der Zertifizierer-ID erstellt wurden!
- Andere Benutzer/Server können die Informationen im Domino-Verzeichnis nutzen, um die Echtheit des vorgezeigten Zertifikats zu prüfen (Authorisierung) und dem Benutzer/Server vertrauen.
- CA-Prozess entkoppelt Erstellung der Benutzer von Signatur (und speichert die Zertifikate zusätzlich zur ID in `admin4.nsf` statt `certlog.nsf`)



Notes-Gegenzertifikate

- Innerhalb einer Organisation (Notes-Domäne) gibt es das gemeinsame Domino-Verzeichnis, was ist aber mit fremden Notes-Benutzern und -Servern?
- Dafür gibt es Notes-Gegenzertifikate!
- Der Name und der öffentliche Schlüssel eines fremden Zertifizierers, Servers oder Benutzers werden mit einer eigenen (OU-)Zertifizierer-ID (oder Server-ID) signiert und in das Domino-Verzeichnis eingetragen.
- Bei der Authentifizierung fremder Benutzer und Server werden dann die (überprüfbaren) Informationen aus dem Gegenzertifikat-Dokument im Domino-Verzeichnis verwendet.



ID-Dateien

- ID-Dateien eines Benutzers enthalten (u. a.)
 - Namen des Besitzers
 - Zertifikat einer (OU-)Zertifizierungs-ID
 - öffentlichen Schlüssel
 - privaten Schlüssel
 - ggf. Internet-Zertifikate (für SSL und S/MIME)
 - ggf. geheime Verschlüsselungsschlüssel (heißt nun mal so ;-)
- Aus dem vergebenen Kennwort wird ein Schlüssel berechnet, mit dem die privaten Daten in der ID-Datei symmetrisch verschlüsselt gespeichert werden. So kann man selbst wenn man die ID-Datei hat ohne das Kennwort nicht an diese Informationen kommen!



Passwort-Wiederherstellung

- In ID-Dateien können Wiederherstellungsinformationen gespeichert werden, mit deren Hilfe die privaten Angaben aus der ID entschlüsselt werden können.
- Diese Informationen werden verschlüsselt gespeichert, so dass normalerweise nur mehrere Administratoren zusammen die ID wiederherstellen können.
- Backups von den ID-Dateien werden dann – ebenfalls verschlüsselt – an eine bestimmte Mail- oder Mail-In-Datenbank geschickt. Diese Backups können im Fall des Verlusts oder der Beschädigung der ID-Datei verwendet werden, um eine neue ID-Datei für den Benutzer zu erstellen.



Anmeldung am Server

- Bei der Anmeldung am Server werden 2 Prüfungen vorgenommen:
 - Validierung des öffentlichen Schlüssels: mit Hilfe des Zertifikats wird der öffentliche Schlüssel aus der ID-Datei geprüft
 - gegenseitige Authentifizierung mit Challenge/Response-Verfahren:
 1. Server erzeugt Zufallszahl, verschlüsselt sie mit öffentlichem Schlüssel des Benutzers und überträgt das Ergebnis
 2. Benutzer entschlüsselt die Zahl und überträgt sie mit dem öffentlichen Schlüssel des Servers verschlüsselt zurück
 3. Der Server entschlüsselt sie und vergleicht sie mit der ursprünglichen Zufallszahl: Stimmen die Zahlen überein, muss es der richtige Benutzer sein.und danach noch einmal andersherum



Organisatorisches

Pragmatische Einführung in Kryptographie

Zertifikate und ID-Dateien

Verschlüsselung und Signaturen

Internet, Protokolle & Server-Zugriff

Sonstiges



Verschlüsselung

- In Notes/Domino gibt es viele Ebenen, auf denen verschlüsselt werden kann:
 - Netzwerk
 - Datenbanken
 - ein- und ausgehende E-Mails
 - Felder in Dokumenten



Netzwerkverschlüsselung

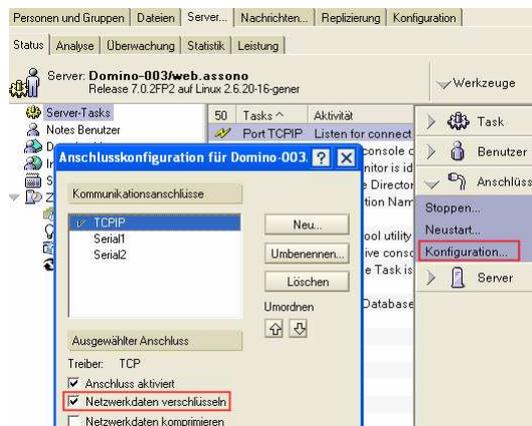
- Der Netzwerk-Verkehr kann bei Notes/Domino verschlüsselt werden.
- Wenn mindestens eine Seite – der Client oder der Server – verschlüsseln möchte, wird verschlüsselt.
- Verschlüsselung wird je Port konfiguriert:

Client:



Netzwerkverschlüsselung (forts.)

Server:





Verschlüsselung von Datenbanken

- Notes-Datenbanken, also nsf- und ntf-Dateien, können verschlüsselt auf dem Datenträger gespeichert werden.
- Bei Server wird mit dessen öffentlichen Schlüssel verschlüsselt, beim Notes-Client mit dem öffentlichen Schlüssel aus der ID-Datei des aktuellen Benutzers. So kann nur der Server bzw. Benutzer die Datenbank mit dem jeweiligen privaten Schlüssel entschlüsseln.
- Es gibt drei Stufen:
Umso höher die Verschlüsselung, desto sicherer ist sie, aber auch langsamer
- Entweder beim Anlegen der Datenbank oder einer Replik gleich richtig einstellen oder nach einer Änderung die Datenbank komprimieren (compact)



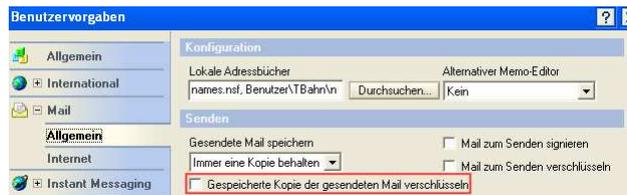
E-Mails (Notes-intern)

- Ein- und ausgehende E-Mails können verschlüsselt werden.
- Die Verschlüsselung eingehender E-Mails wird im Personen-Dokument konfiguriert:



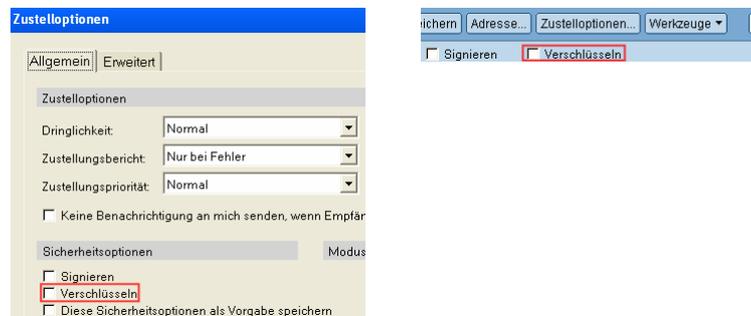
E-Mails (Notes-intern; forts.)

- Beim E-Mail-Versand muss man noch unterscheiden:
 1. Die Verschlüsselung einer ggf. beim Absender gespeicherten Kopie der E-Mail wird in den Benutzervorgaben eingestellt:



E-Mails (Notes-intern; forts.)

2. Die Verschlüsselung versendeter E-Mails kann der Benutzer selbst bestimmen, entweder in den Zustelloptionen oder direkt in der Maske unter der Aktionsleiste:





Verschlüsselung von Feldern in Dokumenten

- Entwickler können für jedes Feld einer Maske einstellen, dass es verschlüsselt werden soll: Feld-Eigenschaften – Erweitert – Sicherheitsoptionen auf „Verschlüsselung für dieses Feld aktivieren“
- Bei RichText-Feldern werden auch die angehängten Dateien verschlüsselt gespeichert. Bei Kennwort-Feldern wird die Option automatisch gesetzt.
- Zusätzlich muss ein Schlüssel festgelegt werden (oder mehrere).
- Dafür gibt zwei Möglichkeiten:
 - öffentliche Schlüssel von Benutzern
 - geheime Verschlüsselungsschlüssel



Verschlüsselung von Feldern in Dokumenten (forts.)

1. öffentliche Schlüssel von Benutzern
 - Es muss ein Feld PublicEncryptionKeys geben, in das die NotesNamen der Personen eingetragen werden müssen, für die das Dokument lesbar sein soll.
 - Beim Speichern oder Senden des Dokuments werden über die NotesNamen die Personen-Dokumente im Domino-Verzeichnis gesucht und der öffentliche Schlüssel der Benutzer ermittelt.
 - Dann werden die gekennzeichneten Felder gegen diese Schlüssel verschlüsselt.



Verschlüsselung von Feldern in Dokumenten (forts.)

2. geheime Verschlüsselungsschlüssel

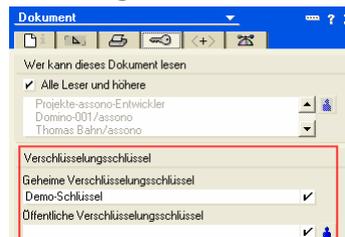
- Über die Sicherheitsoptionen – Notes-Daten – Dokumente können sog. Geheimschlüssel erstellt werden.
- Diese werden in der ID-Datei gespeichert.
- Sie können exportiert oder per E-Mail versendet und dann in andere ID-Dateien importiert werden.



Verschlüsselung von Feldern in Dokumenten (forts.)

2. geheime Verschlüsselungsschlüssel (forts.)

- In den Masken-Eigenschaften kann ein Entwickler unter Sicherheit einen Standard-Verschlüsselungsschlüssel aus den Geheimschlüsseln in seiner ID-Datei auswählen.
- Enthält ein Dokument ein Feld SecretEncryptionKeys, dann erwartet Notes darin den oder die Namen von geheimen Verschlüsselungsschlüsseln.
- In den Dokument-Eigenschaften – Sicherheit kann man für jedes Dokument einzeln geheime oder öffentliche Verschlüsselungsschlüssel (in Form von Personen) auswählen.





Signaturen

- Auch Signaturen gibt es bei Notes/Domino auf verschiedenen Ebenen:
 - ausgehende E-Mails
 - Dokumente
 - Abschnitte



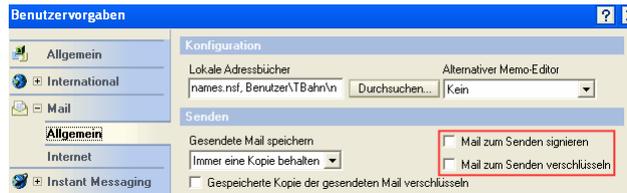
Ausgehende E-Mails (Notes-intern)

- Der Absender kann wie bei der Verschlüsselung selbst bestimmen, ob eine E-Mail signiert werden soll. Dies passiert wieder entweder in den Zustelloptionen oder direkt in der Aktionsleiste:



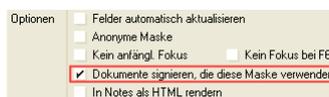
Ausgehende E-Mails (Notes-intern)

- Und wie bei der Verschlüsselung kann man in den Benutzeroptionen die Vorgabe für das Signieren einstellen:



Dokumente

- Ein Entwickler kann in den Masken-Eigenschaften einstellen, dass alle Dokumente, die mit dieser Maske gespeichert oder versendet werden, automatisch mit der aktuellen Benutzer-ID signiert werden:



- Wird so ein Dokument geöffnet, stehen in der Statuszeile die Details der Signatur:

Signiert durch Thomas Bahn/assono am 10.09.2007 01:23:36, gemäß /assono



SSL + HTTP = HTTPS

- Secure Sockets Layer (SSL) oder auch Transport Layer Security (TLS) ist ein Netzwerkprotokoll zur sicheren Übertragung von Daten über das Internet
- SSL zusammen mit Hypertext Transfer Protocol (HTTP) nennt man HTTPS und dient zur Absicherung von Web-Anwendungen durch Verschlüsselung
- HTTPS funktioniert ähnlich wie bei Notes/Domino:
- Server hat ein Zertifikat, das von einer CA signiert wurde
- CA: entweder eine bezahlte Institution oder Gesellschaft (Trustcenter) oder man selbst
- Browser kennen einige wichtige Trustcenter, deren Zertifikate schon vom Browser-Hersteller importiert wurden, aber nicht die eigene CA



SSL + HTTP = HTTPS (forts.)

- Daher vertraut der Browser den selbsterstellten Zertifikaten zunächst nicht. Beim Zugriff auf Web-Server wird man gewarnt und gefragt, ob man dem Zertifikat vertraut.
- Man kann das Zertifikat der eigenen CA in den Browser importieren. Dann vertraut er auch damit signierten Server-Zertifikaten.
- Bei der Anmeldung am Web-Server über HTTPS ist ähnlich wie die erste Hälfte der Anmeldung am Domino-Server
- Danach „weiß“ der Browser, dass der Server der ist, der er vorgibt zu sein und der Netzwerkverkehr passiert danach symmetrisch mit einem Zufallsschlüssel verschlüsselt übertragen.



SSL-Client-Zertifikate

- Zusätzlich kann man bei SSL auch Zertifikate für Benutzer ausstellen: SSL-Client-Zertifikate.
- Diese entsprechen den Zertifikaten in den Notes-ID-Dateien.
- Web-Server können so konfiguriert werden, dass die Benutzer sich nicht per Benutzername und Passwort authentifizieren, sondern ein (meist mit einem Passwort geschützten) SSL-Client-Zertifikat vorweisen müssen.
- Sicheres Verfahren (wie bei Notes), weil so die Kenntnis des Passworts allein nicht reicht: Zwei-Faktor-Authentifizierung



Sichere E-Mails im Internet: S/MIME

- MIME (Multipurpose Internet Mail Extensions): Standard für den Aufbau und die Kodierung von E-Mails im Internet
- MIME-kodierte E-Mails können mehrere Abschnitte enthalten für zum Beispiel Text- und HTML-Version des E-Mail-Bodys und weitere für jeden Anhang und jede eingebettete Grafik
- Domino verschickt E-Mails ins Internet MIME-kodiert
- S/MIME (Secure/Multipurpose Internet Mail Extensions): Standard für die Verschlüsselung und Signatur von E-Mails im Internet
- Funktioniert ähnlich wie Notes-interne Verschlüsselung ;-)
- MIME-Abschnitte werden verschlüsselt mit dem öffentlichen Schlüssel des Empfängers und signiert mit dem privaten Schlüssel des Absenders.
- Signierte E-Mails enthalten auch das Zertifikat des Absenders.



Internet-Gegenzertifikate

- Um die Signatur einer empfangenen S/MIME-E-Mail überprüfen zu können, benötigt man den öffentlichen Schlüssel des Absenders.
- Dazu muss man das Zertifikat des Absenders einmalig in sein persönliches Adressbuch importieren.
- Öffnet man eine signierte E-Mail eines Absenders, dessen Zertifikat man noch nicht importiert hat, bietet der Notes-Client an, das mitgesendete Zertifikat gegenzuzertifizieren.
- Aus Sicherheitsgründen sollte man vor dem Import den Fingerabdruck telefonisch prüfen.



Internet-Gegenzertifikate (forts.)

- Öffnet man danach eine signierte E-Mail dieses Absenders, so kann der Notes-Client die Signatur mit Hilfe des Internet-Gegenzertifikats überprüfen.
- Das Ergebnis dieser Prüfung zeigt er wie bei Notes-internen E-Mails in der Statuszeile an:

Signiert durch Thomas Bahn <tbahn@web.de> am 10.09.2007 21:26:15, gemäß Thomas Bahn/assono



Organisatorisches

Pragmatische Einführung in Kryptographie

Zertifikate und ID-Dateien

Verschlüsselung und Signaturen

Internet, Protokolle & Server-Zugriff

Sonstiges



Quellen

- Administrator-Hilfe: enthält viele Schritt-für-Schritt-Anleitungen und Erklärungen
- [Ebel2004]: „Lotus Notes Domino Administration – Lotus Groupware verwalten, Versionen 5 bis 6.5“, Nadin Ebel, Addison-Wesley, München, 2004
- IBM Redbooks und Redpapers (<http://www.redbooks.ibm.com>):
 - Lotus Security Handbook (SG24-7017-00)
 - Security Considerations in Notes and Domino 7 – Making Great Security Easier to Implement (SG24-7256-00)
 - Domino Designer 6 - A Developer's Handbook (SG24-6854-00)
 - Domino Certification Authority and SSL Certificates
 - Lotus Notes and Domino R5.0 Security Infrastructure Revealed (SG24-5341-00)



Zum guten Schluss...

- Fragen?
 - jetzt stellen oder später
 - E-Mail: tbahn@assono.de
 - Telefon: 04307 900-401
- Folien und Passwort-Safe unter <http://www.assono.de/blog.nsf/d6plinks/AdminCamp2007>
- In eigener Sache – wir suchen Verstärkung:
IT-Berater (m/w)
Details unter <http://www.assono.de/jobs.html>