Bankers Campus
ERFOLGREICH AUF DEN PUNKT

assono

Anwendertreffen IBM Notes & Domino 2016:

Weil sicher sicher sicher ist! –
Admin-Tipps für ein sicheres Notes und Domino

Potsdam, 26. April 2016

Innovative Software-Lösungen.

www.assono.de

Thomas Bahn

Diplom-Mathematiker, Universität Hannover

seit 1997 entwickle ich mit Java und relationalen Datenbanken

seit 1999 mit Notes/Domino zu tun: Entwicklung, Administration, Beratung und Schulungen

regelmäßiger Sprecher auf nationalen und internationalen Fachkonferenzen zu IBM Lotus Notes/Domino und Autor für THE VIEW

tbahn@assono.de
http://www.assono.de/blog
04307/900-401

assono
IT-Consulting & Solutions

Agenda

- Berechtigungen – vom Netzwerk bis zum Abschnitt
- Zertifikate und ID-Dateien
- Verschlüsselung – vom Netzwerk bis zum Feld
- Signaturen
- die Antworten auf „gestern“
  - Execution Control List
  - Code in gespeicherten Masken
  - Zugriff von außen, z. B. aus Office-Makros
- Neuerungen in 2015/2016
- zum später Nachlesen:
  - Pragmatische Einführung in die Kryptographie
  - Authentifizierung
  - Passwort-Management
  - Sicherheit in Notes-Anwendungen

Worum geht es in dieser Session nicht?

Worum geht es in dieser Session **nicht**?

- Safety vs. Security
- Datensicherung, Verfügbarkeit
- Sicherheit von Notes-Client-Plug-Ins, Widgets, Composite Apps
- physische Sicherheit
- Betriebssystem, Dateisystem, Netzwerk, Anti-Virus…
- weiter-/tiefergehende Details

Berechtigungen – vom Netzwerk bis zum Abschnitt

Berechtigungsebenen

- Netzwerk
- Server
- Verzeichnis
- Datenbank
- Dokument
- Kontrollierter Zugriff-Abschnitt

Netzwerk

- Allow_Access_<*port*> in notes.ini
  www-10.lotus.com/ldd/dominowiki.nsf/dx/allow_access_portname
- Deny_Access_<*port*> in notes.ini
  www-10.lotus.com/ldd/dominowiki.nsf/dx/deny_access_port
- Beispiele:
  Allow_Access_ClusterTCPIP=ClusterServerUndAdmins
  Deny_Access_TCPIP=AusgeschiedeneMitarbeiter
- Verteilung z. B. über das *-Konfigurationsdokument

Server-Dokument – Sicherheit – Administratoren

- Administratoren mit voller Berechtigung:
  - nur Personen
  - Ereignis-Generator und -Handler in events4.nsf einrichten für eine Benachrichtigung, jedes Mal wenn aktiviert
  - hilft nicht gegen Verschlüsselung
- Administratoren: LocalDomainAdmins
- weitere Felder werden eher selten verwendet
- Administratoren mit voller Remotekonsolen-Berechtigung:

  LocalDomainServers oder Agent-Signer-IDs, wenn Agenten Serverkonsolen-Kommandos nutzen können sollen (NotesSession.SendConsoleCommand)

Server-Dokument – Sicherheit – Programmierbarkeit

- Wer darf Code ausführen
  - Formelsprache
  - Beschränktes LotusScript/Java
  - Unbeschränktes LotusScript/Java (Vorsicht: **OS-Befehle!**)
- „Agenten signieren, die im Namen anderer ausgeführt werden", „Agenten oder XPages signieren, die im Namen des Aufrufers ausgeführt werden", „Scriptbibliotheken signieren, die im Namen anderer ausgeführt werden" und ggf. „Unbeschränkte Methoden und Operationen signieren oder ausführen":
  - nur Personen

Server-Dokument – Sicherheit – Sicherheitseinstellungen

- Public Key Checking
  - Öffentliche Schlüssel vergleichen: Überprüfen von öffentlichen Schlüsseln für in vertrauenswürdigen Verzeichnissen aufgeführte Notes-Benutzer und Domino-Server zwingend
  - Nichtübereinstimmungen von öffentlichen Schlüsseln protokollieren: Nichtübereinstimmungen von öffentlichen Schlüsseln für alle Notes-Benutzer und Domino-Server protokollieren
  - mindestens protokollieren, um zu korrigieren
  - sonst kann Benutzer für ihn verschlüsselte Informationen (wie E-Mails) nicht lesen
- Anonyme Notes Verbindungen zulassen: Nein
- Kennwörter von Notes-IDs überprüfen: Aktiviert

Server-Dokument – Sicherheit – Internetzugriff

- Internet-Authentifizierung:
  Weniger Namensvariationen mit höherer Sicherheit

Server-Dokument – Sicherheit – Auf Server zugreifen

- Serverzugriff:

  In allen vertrauenswürdigen Verzeichnissen aufgeführte Benutzer und Server-Name und LocalDomainServers

- Kein Serverzugriff:

  Gruppe für ausgeschiedene Mitarbeiter und Praktikanten

Server-Dokument – Sicherheit – Auf Server zugreifen

- Datenbanken und Schablonen erstellen:
  LocalDomainAdmins, LocalDomainServers
- Neue Repliken erstellen:
  LocalDomainAdmins, LocalDomainServers
- Masterschablonen erstellen:
  LocalDomainAdmins, LocalDomainServers
- Verwendung von Monitoren zulässig für:
  LocalDomainAdmins
- Vertrauenswürdige Server (damit Agenten auf andere Server zugreifen können):
  LocalDomainServers

Server-Dokument – Ports… – Internet-Ports

- SSL-Verschlüsselungscodes:
  - ~~AES aktivieren, alles mit MD5 MAC und DES deaktivieren, ggf. Triple-DES und RC4 deaktivieren~~
  - über notes.ini (SSLCipherSpec) steuern
- Einstellungen zum Serverzugriff erzwingen:
  - Ja (für alle Protokolle), sonst funktioniert “Kein Serverzugriff” nicht für die Internet-Protokolle
- Alle Ports deaktivieren, die nicht benötigt werden – verhindert Prozessstart „aus Versehen“

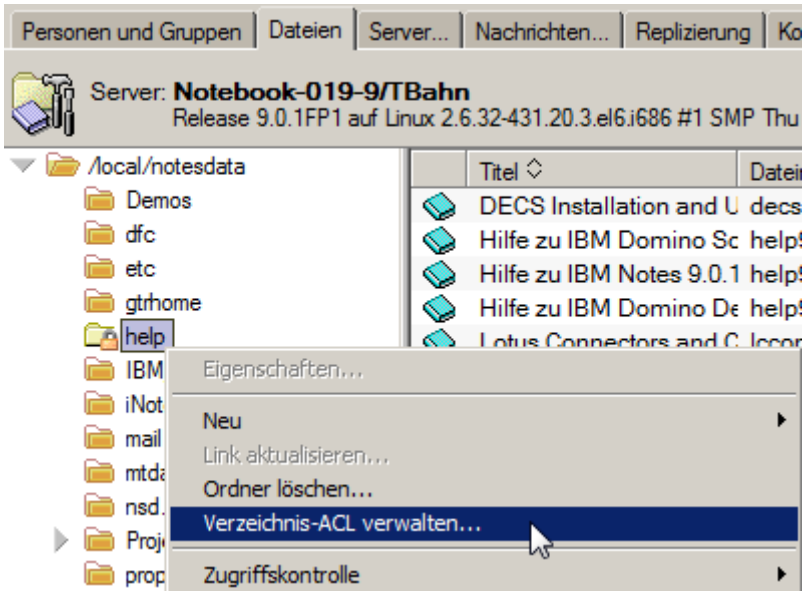
Server-Dokument – Notes Traveler

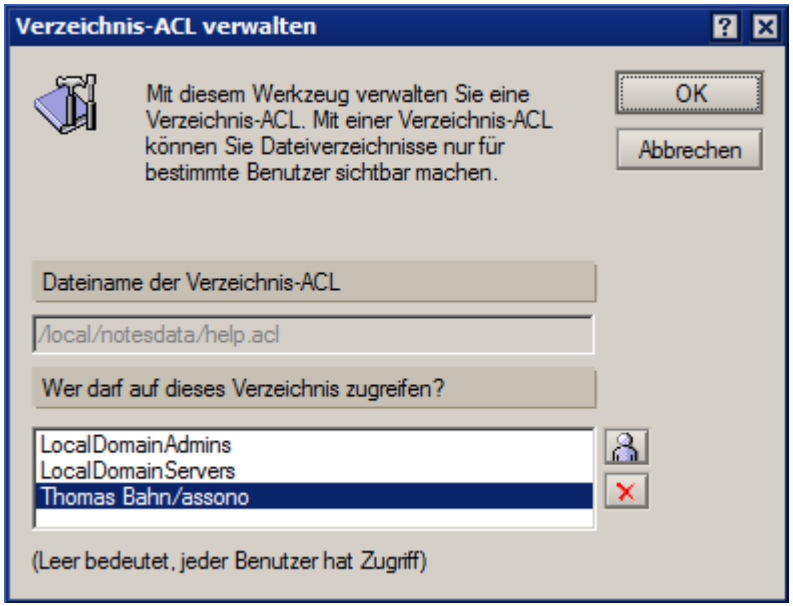
- Zugriff auf IBM Notes Traveler
- Serverzugriff:
  - In allen vertrauenswürdigen Verzeichnissen aufgeführte Benutzer
- Kein Serverzugriff:
  - Gruppe für ausgeschiedene Mitarbeiter und Praktikanten

Verzeichnis

- Voraussetzung: Enable_ACL_Files=1 in notes.ini
  www-10.lotus.com/ldd/dominowiki.nsf/dx/enable_acl_files

- z. B. über *-Konfigurationsdokument setzen





Datenbank

- Zugriffskontrolle = Access Control List = ACL
- 7 Berechtigungsstufen
  - Manager
  - Entwickler
  - Editor
    - nur in Ausnahmefällen sinnvoll
    - Felder konfigurierbar: nur für ≥Editoren änderbar (PROTECTED-Flag)
  - Autor
    - für mich einzige sinnvolle Option für alle Anwender
  - Leser
  - Einlieferer
  - Kein Zugriff
    - Sollwert für -Default- und ggf. Anonymous

Datenbank (forts.)

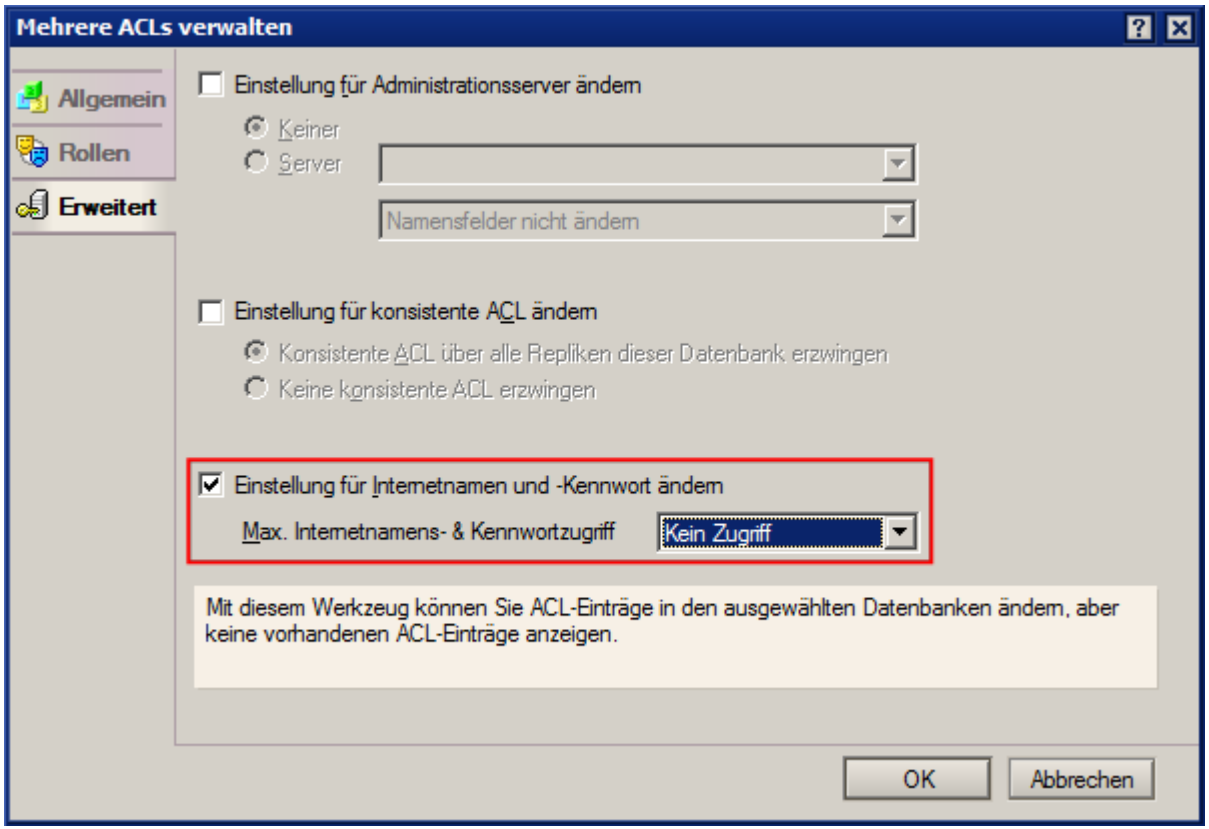
- Weitere Berechtigungen
  - Dokumente erstellen
  - Dokumente löschen
  - Private Agenten erstellen
  - Private Ordner/Ansichten erstellen
    - geht aber immer!
  - Gemeinsame Ordner/Ansichten erstellen
  - LotusScript/Java-Agenten erstellen
  - Öffentliche Dokumente lesen
  - Öffentliche Dokumente schreiben
  - Dokumente replizieren oder kopieren
    - gilt auch innerhalb von Dokumenten (Zwischenablage) und für das Drucken!
- Rollen

Datenbank

- ACL – Erweiterte Einstellungen
  - Administrationsserver
    - **immer** setzen!
  - Konsistente ACL
    - „Nebenwirkung”: Rechte und Rollen wirken auch lokal

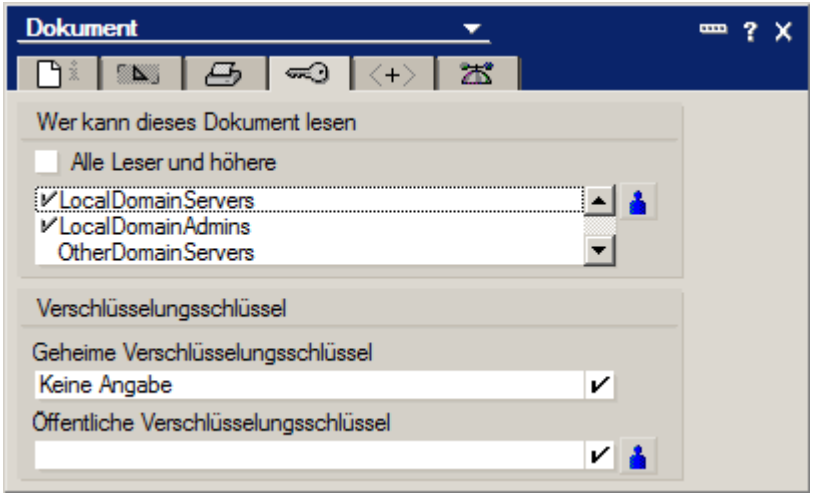
Datenbank

- Maximaler Internetzugriff
  - schneller "Generalschalter" für eine Datenbank
  - lässt sich im Admin-Client einfach für viele Datenbanken gleichzeitig einstellen, z. B. vor erster Aktivierung des HTTP-Tasks



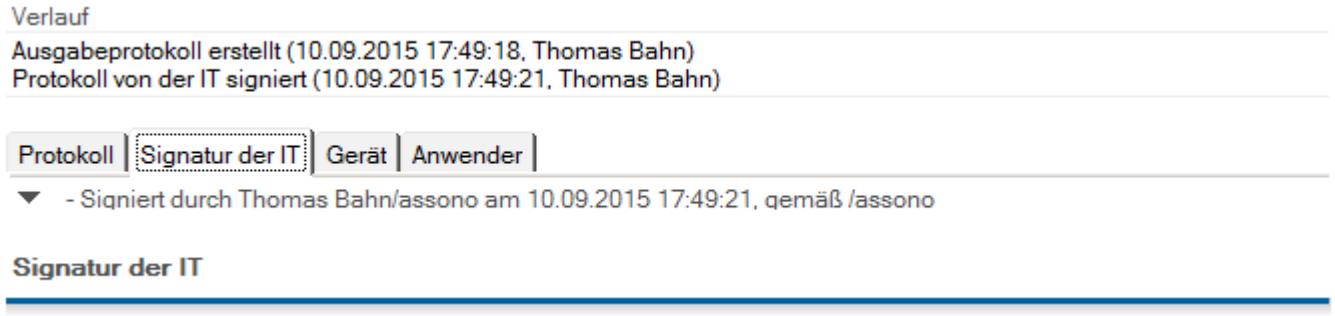
Dokument

- Leser-Felder
  - wenn vorhanden und nicht leer: exklusiv
  - bei mehreren: additiv
- Bearbeiter-Felder
  - zählen auch als Leser-Felder
  - relevant nur für Autoren
  - mehrere wirken additiv
- Dokument-Eigenschaften
  - auch für normale Nutzer
  - getrennt für jedes Dokument
  - gespeichert in $Readers-Item



Kontrollierter Zugriff-Abschnitt

- nur von Entwicklern nutzbar
- beschränken Bearbeiter-Zugriff für Felder eines Abschnitts
- sehr relevant auch für elektronische Signaturen:
  Wer hat die Felder des Abschnitts als letztes verändert?



- Nach Manipulation:

Zertifikate und ID-Dateien

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

- CA-Prozess entkoppelt Erstellung der Benutzer von Signatur (und speichert die Zertifikate zusätzlich zur ID in admin4.nsf statt certlog.nsf)
- ohne CA-Prozess: Zugriff auf Zertifizierer-ID nötig für neue Benutzer und Rezertifizierungen
- eingeschränkte Rechte für Helpdesk-Mitarbeiter
- für ADSync und andere Tools, die neue Notes-Benutzer automatisiert erzeugen sollen

Notes-Gegenzertifikate

- innerhalb einer Organisation (Notes-Domäne) gibt es das gemeinsame Domino-Verzeichnis, was ist aber mit fremden Notes-Benutzern und Domino-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 (heißen 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!

Vergessene Passwörter und verlorene ID-Dateien

- ID-Dateien mit bekannten Passwörtern im Dateisystem aufbewahren (**nicht** machen!)
- Passwortwiederherstellung und Mail-in-DB für ID-Dateien (besser) – seit R5
- ID Vault (am besten) – seit 8.5
- Drittherstellerprodukte

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.

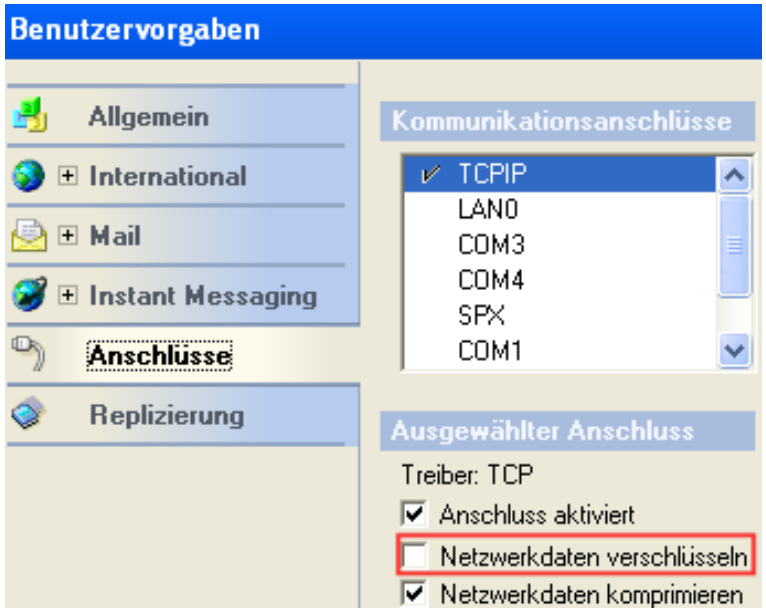
Verschlüsselung – vom Netzwerk bis zum Feld

Verschlüsselung

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

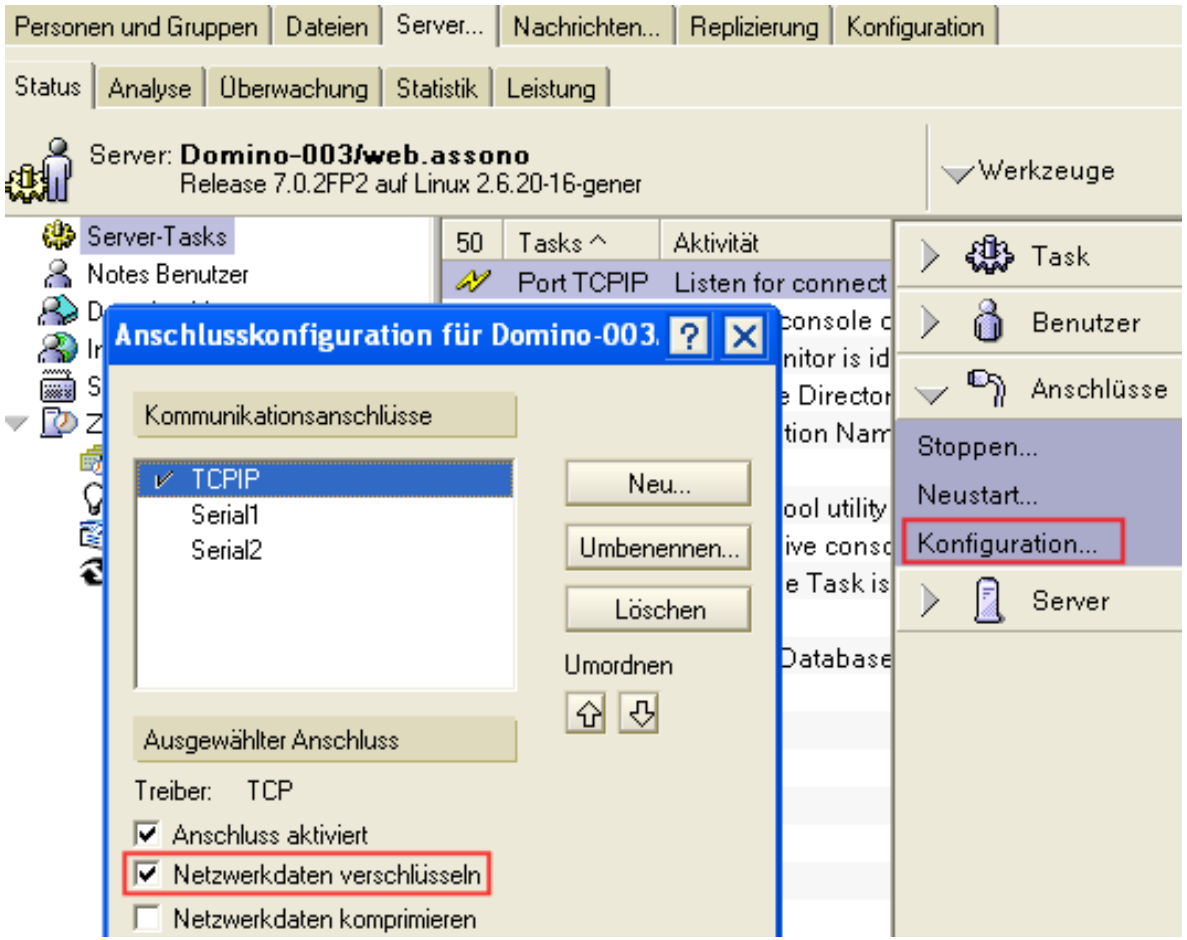
Netzwerkverschlüsselung

- 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:



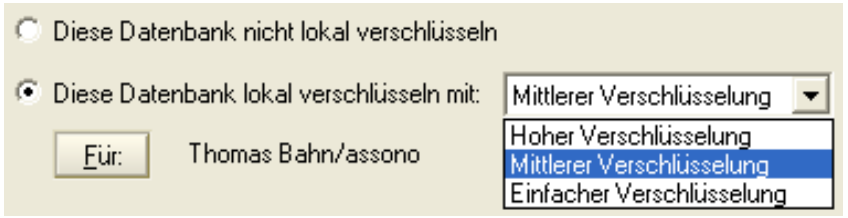
Netzwerkverschlüsselung (forts.)

- Server:



Verschlüsselung von Datenbanken

- Notes-Datenbanken und -Schablonen 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.



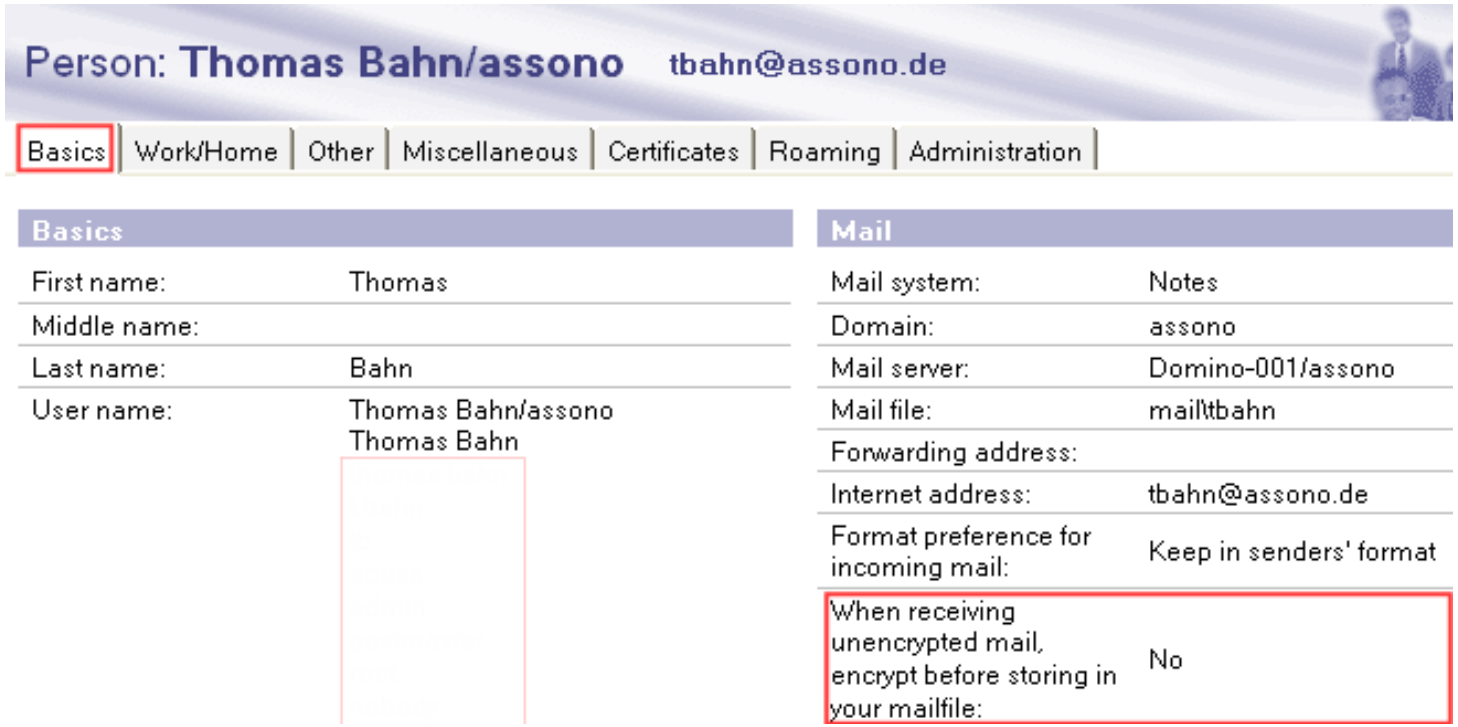
- Es gibt drei Stufen:
- Umso höher die Verschlüsselung, desto sicherer ist sie, aber auch langsamer
- Entweder beim Anlegen der Datenbank oder einer Replik gleich richtig einstellen oder nach einer Änderung die Datenbank komprimieren (compact)
- auf dem Server nur sinnvoll, wenn Server-ID gesichert

nlo-Dateien (DAOS)

- Vorgabe: nlo-Dateien werden mit privaten Schlüssel aus Server-ID verschlüsselt
- sinnvoll überhaupt nur, wenn Server-ID passwort-gesichert
- macht aber einiges umständlicher, z. B.
  - Rücksicherung auf anderen Server
  - Deduplizierung auf SAN
- Ausschalter: DAOS_Encrypt_NLO=0 in notes.ini
- muss konfiguriert werden, **bevor** DAOS auf Server aktiviert wird
- ab 9.0.1 FP3 kann der DAOSMGR auch nachträglich ver-/entschlüsseln: www.ibm.com/support/docview.wss?uid=swg21673931

E-Mails (Notes-intern)

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



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

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

Basics

First name:	Thomas
Middle name:	
Last name:	Bahn
User name:	Thomas Bahn/assono Thomas Bahn

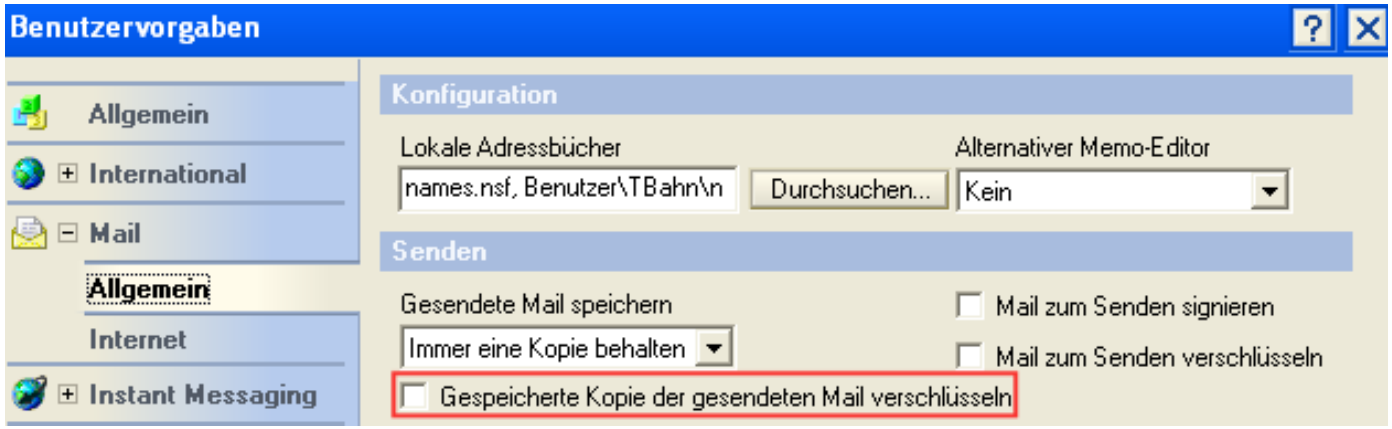
Mail

Mail system:	Notes
Domain:	assono
Mail server:	Domino-001/assono
Mail file:	mail\tbahn
Forwarding address:	
Internet address:	tbahn@assono.de
Format preference for incoming mail:	Keep in senders' format
When receiving unencrypted mail, encrypt before storing in your mailfile:	No

E-Mails (Notes-intern; forts.)

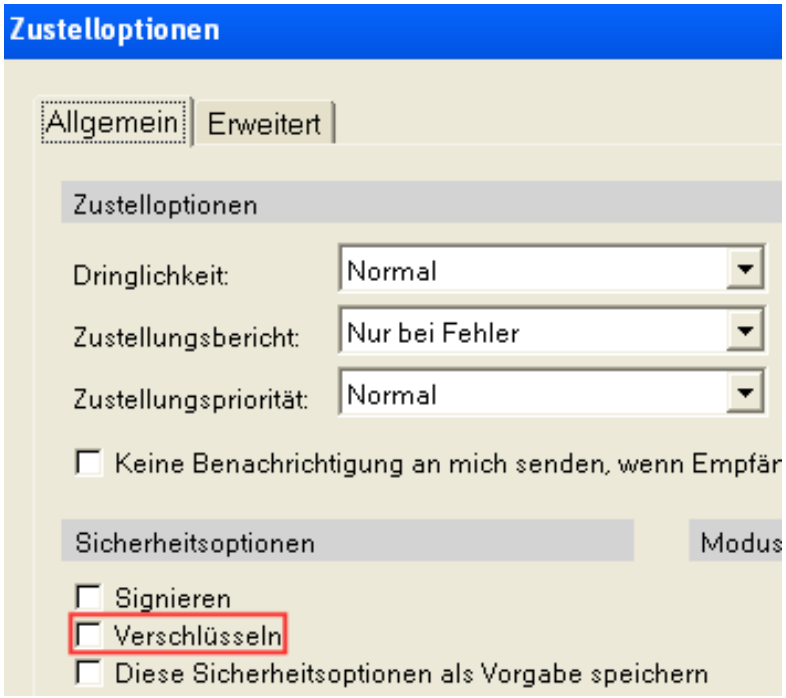
- beim E-Mail-Versand muss man noch unterscheiden:

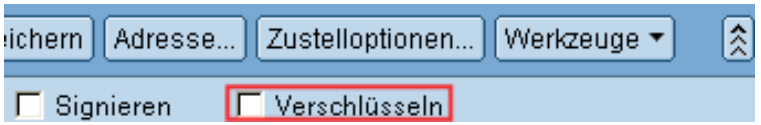
1. Die Verschlüsselung einer ggf. beim Absender gespeicherten Kopie der E-Mail wird in den Benutzervorgaben eingestellt:



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.

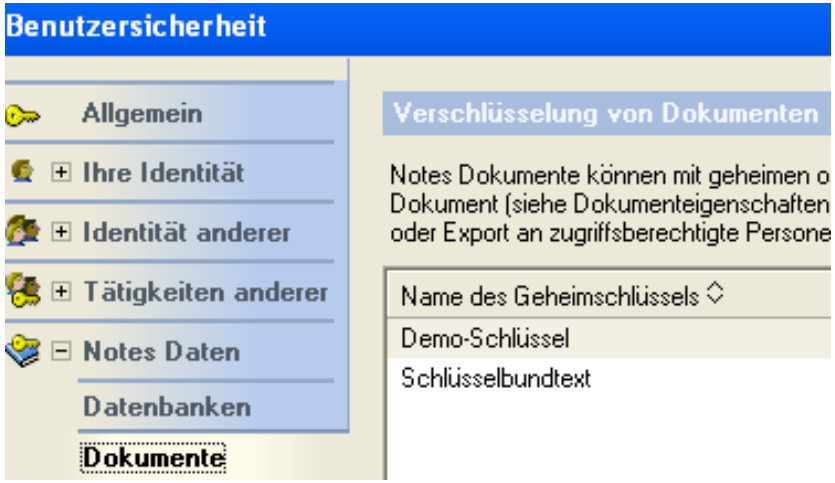
Verschlüsselung von Feldern in Dokumenten (forts.)

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.



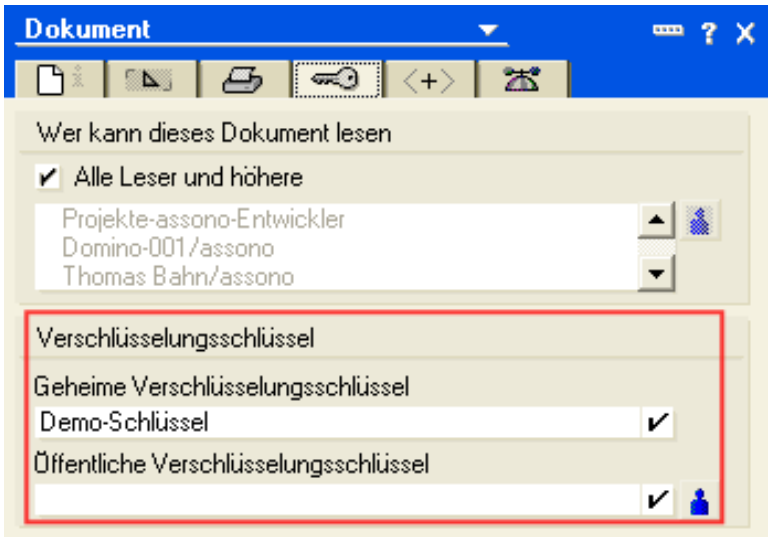
Verschlüsselung von Feldern in Dokumenten (forts.)

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.



Preisfrage

- Was muss man beachten, wenn man mit NotesDocument.Save (LotusScript) ein Dokument mit zu verschlüsselnden Feldern speichert?

Antwort zur Preisfrage

- Was muss man beachten, wenn man mit NotesDocument.Save (LotusScript) ein Dokument mit zu verschlüsselnden Feldern speichert?

- Man muss vorher NotesDocument.Encrypt aufrufen, sonst werden die Felder **unverschlüsselt** gespeichert!

- Leider funktioniert NotesDocument.Encrypt nicht, solange das Dokument im Client geöffnet ist…

- Problem z. B. für PostSave-Ereignisbehandler

Umstellen auf AES für Notes-Mails und -Dokumente

- Verschlüsselung von Notes-internen E-Mails und Dokumenten normalerweise mit RC2
- Umstellung auf AES
  - Voraussetzung: Benutzer-IDs min. 1024-Bit-RSA-Schlüssel
  - "Sicherheitseinstellungen"-Dokument →
  - Reiter "Schlüssel und Zertifikate" →
  - Abschnitt "Verschlüsselungseinstellungen für Dokumente/Mail" →
  - Feld "Verschlüsselungsanforderungen:" auf "FIPS-140-2-Algorithmen für Notes-Verschlüsselung verwenden (Server und Client ab 8.0.x erforderlich)"
  - Sicherheitseinstellungen per Richtlinie zuweisen
- Eintrag „AES für die Verschlüsselung von Mail und Dokumenten konfigurieren" in der Admin-Hilfe

Unterstützte Algorithmen und Schlüssellängen

- Und welche Algorithmen und Schlüssellängen werden nun unterstützt und verwendet?

- Supported key sizes in Notes/Domino
  www.lotus.com/ldd/dominowiki.nsf/dx/supported-key-sizes-in-notesdomino

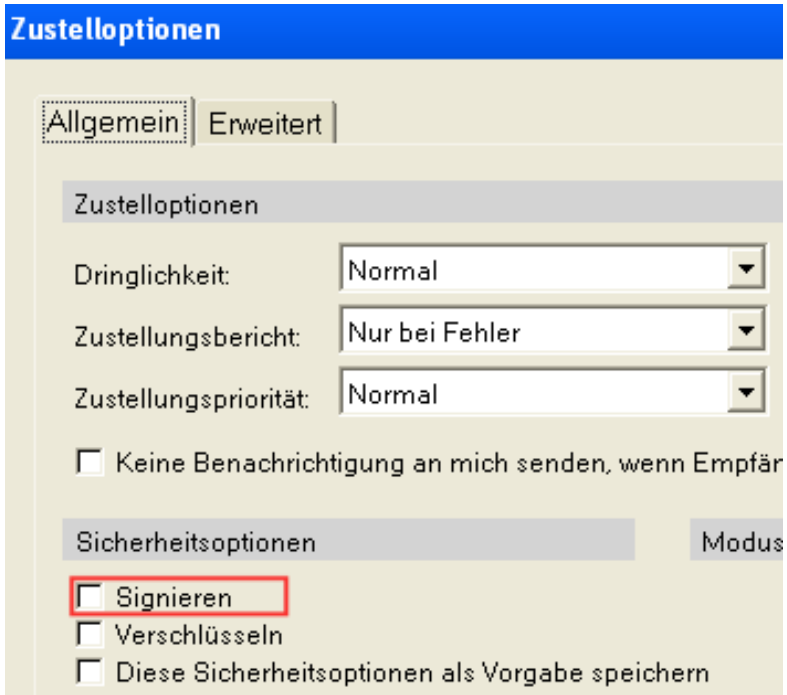
Signaturen

Signaturen

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

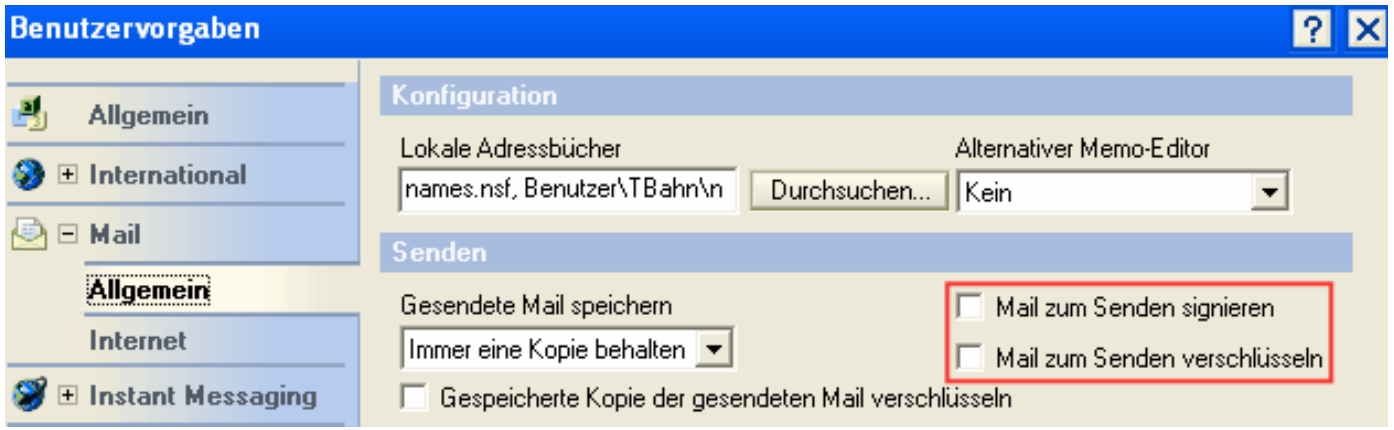
Ausgehende E-Mails (Notes-intern)

- 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 Maske unter der Aktionsleiste:



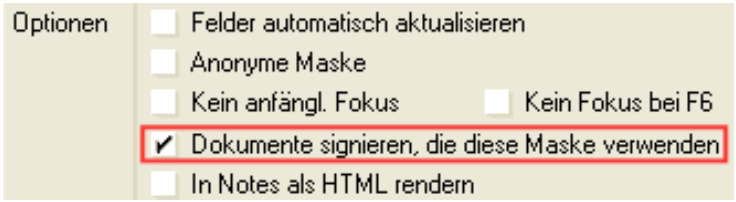
Ausgehende E-Mails (Notes-intern)

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



Dokumente

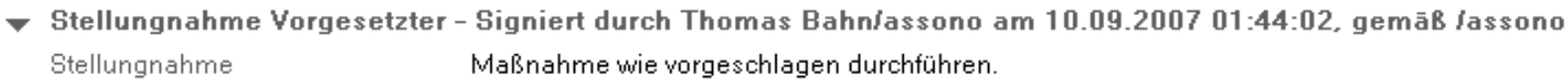
- Ein Entwickler kann in den Masken-Eigenschaften einstellen, dass alle Dokumente, die mit dieser Maske gespeichert oder versendet werden, automatisch mit der aktuellen Benutzer-ID signiert werden:



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

Abschnitte

- Zusätzlich können auch Kontrollierter-Zugriff-Abschnitte signiert werden. Das sieht dann zum Beispiel so aus:



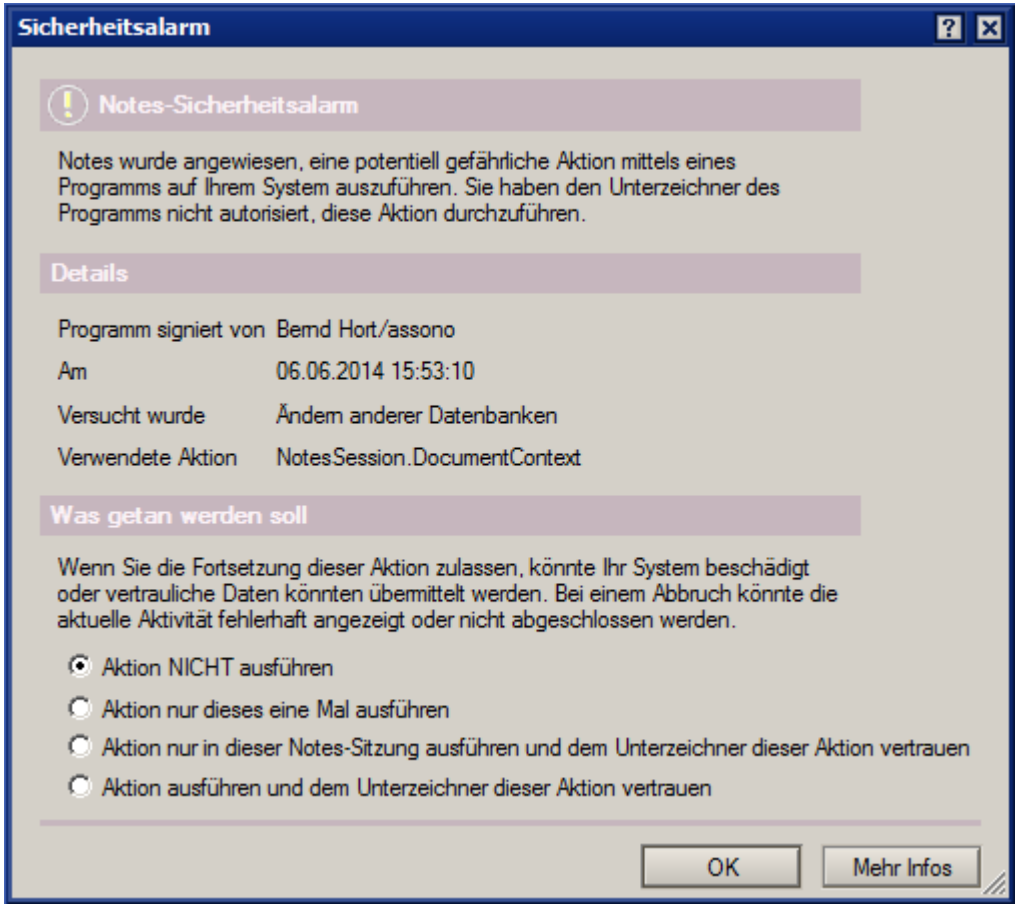
- Dazu muss der Entwickler in den Feldeigenschaften – Erweitert – Sicherheitsoptionen „Signieren beim Versenden oder Speichern im Abschnitt" für mindestens ein Feld innerhalb des Abschnitts auswählen.

- Beim Speichern oder Senden eines Dokuments wird dann die Signatur jedes Abschnitts aktualisiert, den der aktuelle Benutzer bearbeiten darf.

Execution Control List (ECL)

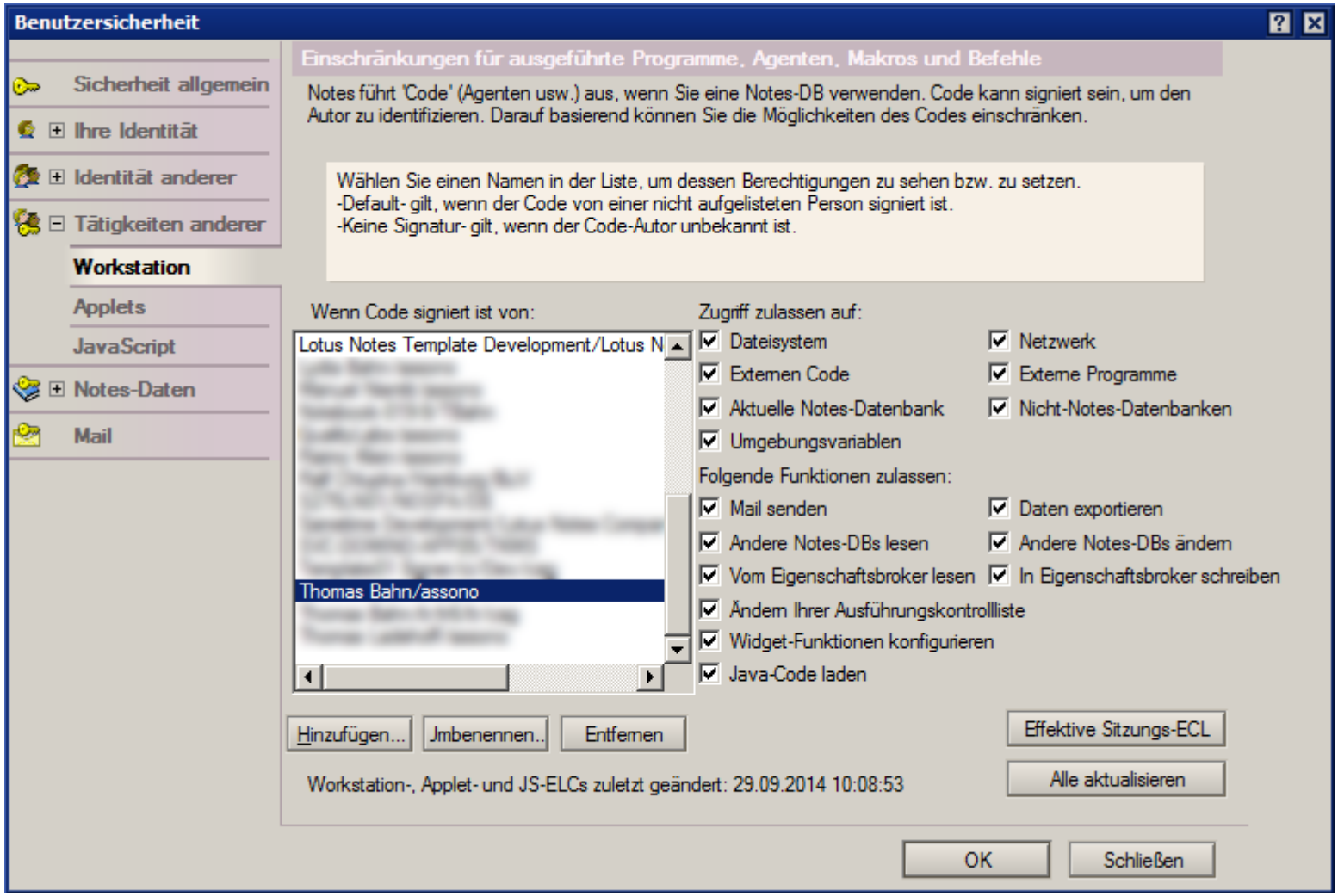
Execution Control List (ECL)

- ECL legt fest, wer „gefährlichen" Code ausführen darf
- Code ist grundsätzlich signiert
- bei potentiell gefährlichen Befehlen wird in der ECL geprüft, ob der Signierer des Codes diesen Befehl ausführen darf
- falls nicht, wird der Benutzer gefragt:



Execution Control List (ECL; forts.)

- Datei – Sicherheit – Benutzersicherheit…



Execution Control List (ECL; forts.)

- per Sicherheitseinstellung und Richtlinie administrativ steuerbar:

Execution Control List (ECL; forts.)

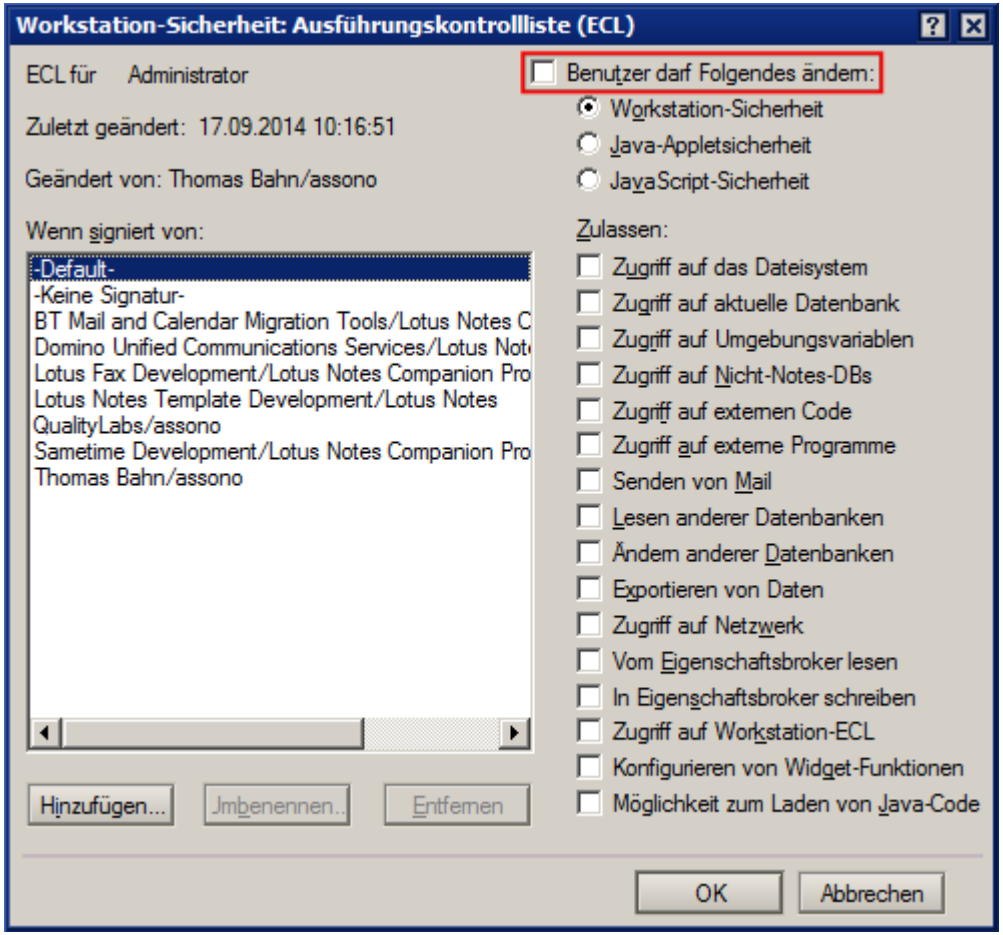
- … und was ist jetzt „Best Practice“?
- bewährte Lösung:
  - persönliche Signier-ID für jeden Berechtigten
    - mit dieser ID wird **nicht** normal gearbeitet, sie ist ausschließlich nur für das Signieren da
  - oder eine Signier-ID, aber mit mehreren Passwörtern (4-Augen-Prinzip)
- Variante: mehrere Signier-IDs mit unterschiedlichen Berechtigungen in der ECL

Execution Control List (ECL; forts.)

- Vorteile:
  - Nachvollziehbarkeit: Wer war's?
  - Einfachheit: nur begrenzte Anzahl von Einträgen in ECL
  - Fälschungssicherheit:
    mit Server-ID unterzeichnen per Admin-Prozess (z. B. aus LotusScript) ergibt Warnungen bei Ausführung
- Vorgehen
  - **immer** vor dem Einspielen einer neuen Version die Schablone signieren

Execution Control List (ECL; forts.)

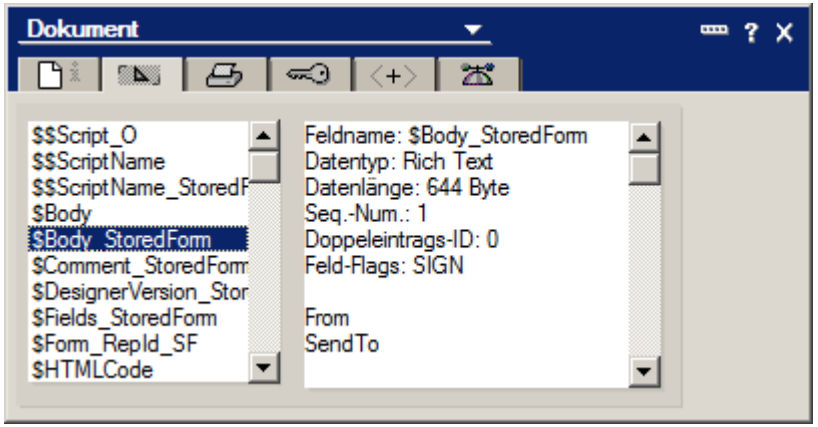
- nur die Signier-IDs werden in die ECL eingetragen (ggf. zusätzlich zu Lotus/IBM-Einträgen)
- **nicht** die Server-IDs
- -Default- und -Keine Signatur- dürfen nichts
- Benutzern das Ändern der ECL verbieten:



Code in gespeicherten Masken

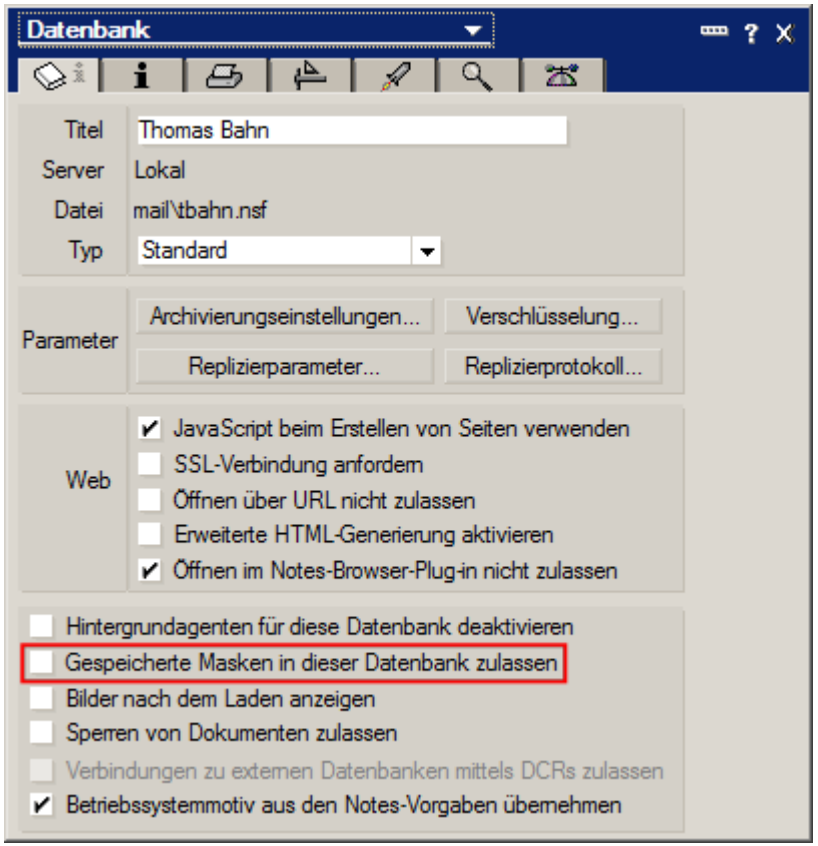
Code in gespeicherten Masken

- E-Mails können auch die ursprünglichen Masken „transportieren“
- Dokumente werden dann in der „mitgebrachten“ Maske geöffnet statt der Standard-Nachrichten-Maske (Memo)
- zu erkennen an den $..._StoredForm-Items:



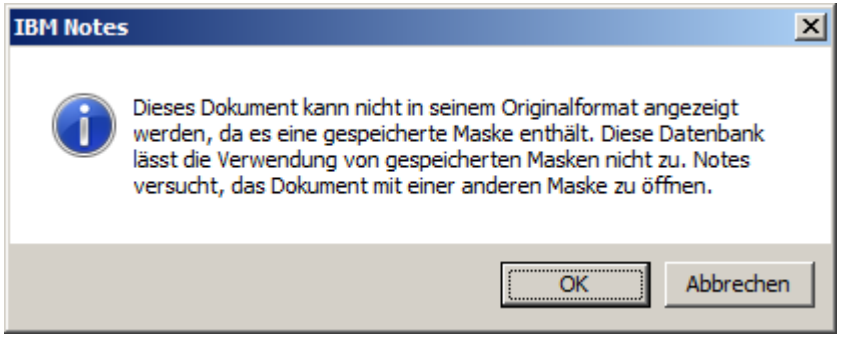
Code in gespeicherten Masken (forts.)

- Beim Öffnen wird auch Code z. B. in den Ereignis-Behandlern QueryOpen, PostOpen usw. ausgeführt
- lässt sich einfach verhindern:



Code in gespeicherten Masken (forts.)

- Beim Öffnen des Dokuments kommt dann:

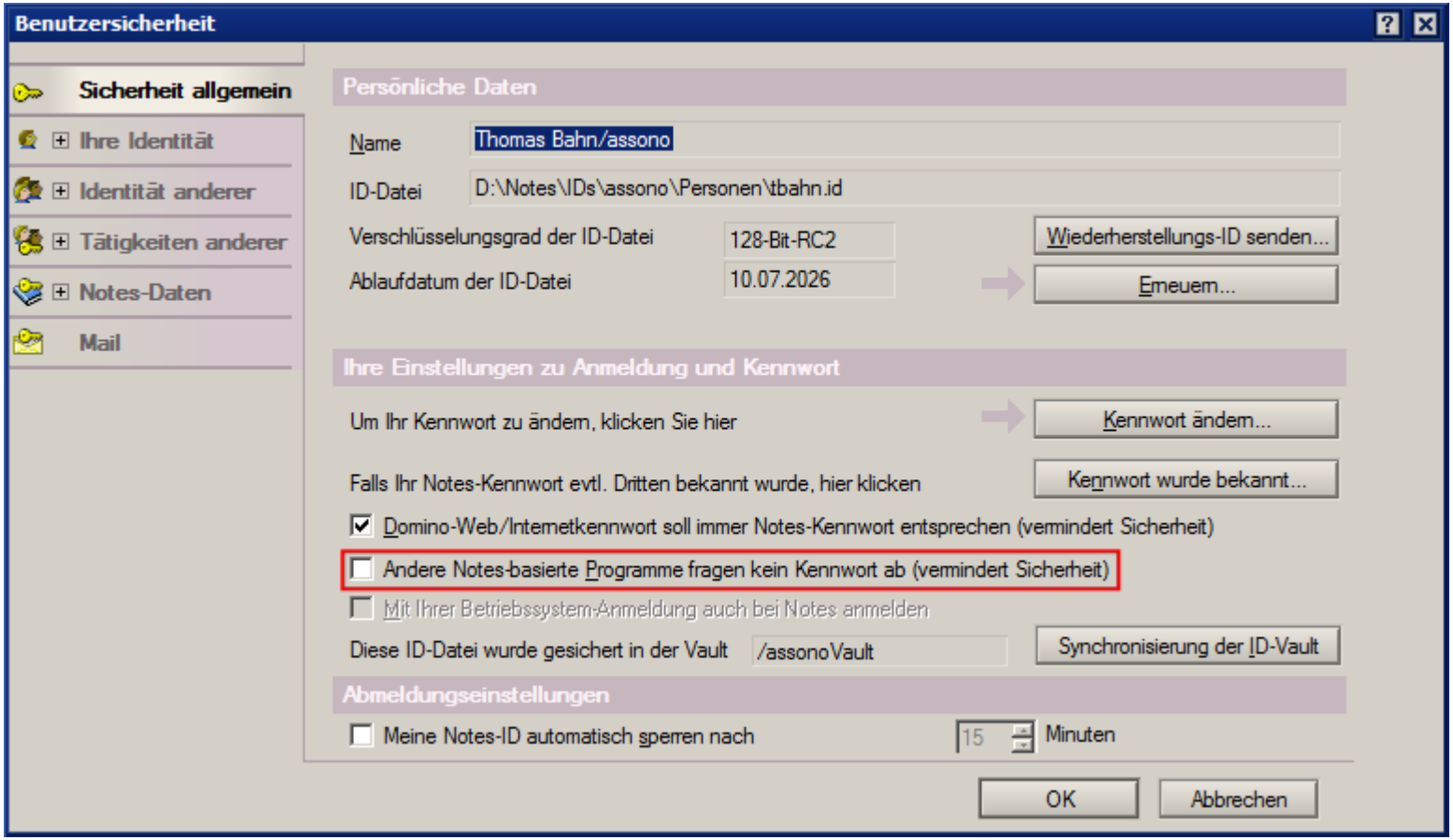


- Verteilung:
  einfach Schablone „richtig" einstellen und Designer laufen lassen

Zugriff von außen

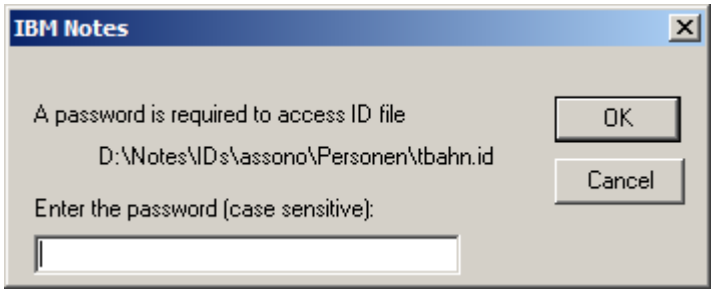
Zugriff von außen

- aus Office-Makros, Scripting Host, … über COM/OLE-Schnittstelle
- Passwort-Eingabe erzwingen:



Zugriff von außen (forts.)

- Passwort-Eingabe sieht anders aus als normal:



Zugriff von außen (forts.)

- per Sicherheitseinstellung und Richtlinie administrativ steuerbar:

Neuerungen 2015/2016

aktuelle Sicherheitsprobleme

- POODLE, erst SSL v3, dann auch bei TLS
- Logjam,
- Bar Mitzvah und
- immer wieder Java, Java, Java, …

POODLE mit SSL

- „SSLv3 contains a vulnerability that has been referred to as the Padding Oracle On Downgraded Legacy Encryption (POODLE) attack, which is a man-in-the-middle attack affecting Web browsers.“
- Problem ist das Verfahren an sich, kein Implementierungsfehler.
- Also: SSLv3 muss abgeschaltet werden!
- How is IBM Domino impacted by the POODLE attack?, www.ibm.com/support/docview.wss?uid=swg21687167

POODLE mit TLS

- Transport Layer Security (TLS) padding vulnerability via a POODLE (Padding Oracle On Downgraded Legacy Encryption) like attack affects IBM Domino.
- Eine Variante von POODLE, die auch TLS betrifft – diesmal aber „nur" die Implementierung, die korrigiert werden kann.
- Security Bulletin: TLS Padding Vulnerability affects IBM Domino (CVE-2014-8730, CVE-2014-3566), www.ibm.com/support/docview.wss?uid=swg21693142

RC4 „Bar Mitzvah"-Angriff auf SSL/TLS

- „The RC4 algorithm, as used in the TLS protocol and SSL protocol, could allow a remote attacker to obtain sensitive information. An attacker could exploit this vulnerability to remotely expose account credentials without requiring an active man-in-the-middle session. […] This vulnerability is commonly referred to as 'Bar Mitzvah Attack'."

- Security Bulletin: Vulnerability in RC4 stream cipher affects IBM Notes, Domino & Expeditor (CVE-2015-2808), www.ibm.com/support/docview.wss?uid=swg21902300

Logjam: Angriff gegen das TLS-Protokoll

- „The Logjam attack allows a man-in-the-middle attacker to downgrade vulnerable TLS connections to 512-bit export-grade cryptography. This allows the attacker to read and modify any data passed over the connection. […] is due to a flaw in the TLS protocol […]“

- Weak Diffie-Hellman and the Logjam Attack
  https://weakdh.org

aktuelle Sicherheitsupdates

- Fix Packs
- Interims Fixes
- JVM-Patches
  - ersetzen nur Java Virtual Machine (JVM)
  - abhängig vom Fix Pack, im gleichen „Rang“ wie Interims Fixes

Domino-Server lernt Neues: TLS 1.0

- 9.0.1 FP2 IF3, 9.0 IF7, 8.5.3 FP6 IF6, 8.5.2 FP4 IF3, 8.5.1 FP5 IF3
- HTTP, SMTP, LDAP, POP3 & IMAP, inbound & outbound
- TLS_FALLBACK_SCSV to prevent the POODLE downgrade attack
- SSLv2 entfernt
- SSL renegotiation has been disabled
- All weak (<128 bits) cipher suites have been disabled
- Prevents both Poodle attacks: CVE-2014-3566 and CVE-2014-8730
- IBM Domino Interim Fixes to support TLS 1.0 which can be used to prevent the POODLE attack, www-10.lotus.com/ldd/dominowiki.nsf/dx/IBM_Domino_TLS_1.0

Domino-Server lernt Neues: SSLv3 deaktivieren

- 9.0.1 FP3, 9.0.1 FP2 IF3, 9.0 IF7, 8.5.3 FP6 IF6, 8.5.2 FP4 IF3, 8.5.1 FP5 IF3
- The notes.ini parameter […] DISABLE_SSLV3=1 allows Domino server to disable SSLv3.
- How to disable SSLv3 on a IBM Domino Server, www.ibm.com/support/docview.wss?uid=swg21695998
- **Vorsicht:** „DIIOP and iSpy do not currently support TLS and will not function if the notes.ini DISABLE_SSLV3=1 is set [...].“
- TLS 1.2 deliveries for IBM Notes & Domino 9.x, www.ibm.com/support/docview.wss?uid=swg21697925

Domino-Server lernt Neues: SHA-2

- Notes & Domino 9.0.1 Fix Pack 3
- „configure Domino 9.x to use a SHA-2 certificate over HTTP, SMTP, LDAP, POP, and IMAP“
- „import a 3rd-party SHA-2 cert or generate SHA-2 certs“
- „we will not be able to support SHA-2 on Domino 8.5.x.“
- SHA-2 support available for IBM Domino 9.x, www.ibm.com/support/docview.wss?uid=swg21418982

Domino-Server lernt Neues: SHA-2 (forts.)

- Generating a SHA-2 Keyring file, www.lotus.com/ldd/dominowiki.nsf/dx/Domino_keyring
- Generating a keyring file with a self-signed SHA-2 cert using OpenSSL and kyrtool, www.lotus.com/ldd/dominowiki.nsf/dx/Self-signed_SHA-2_with_OpenSSL_and_kyrtool
- Generating a keyring file with a third party CA SHA-2 cert using OpenSSL and kyrtool, www.lotus.com/ldd/dominowiki.nsf/dx/3rd_Party_SHA-2_with_OpenSSL_and_kyrtool
- Installing and Running the Domino keyring tool, www.lotus.com/ldd/dominowiki.nsf/dx/kyrtool

Domino-Server lernt Neues: TLS 1.2

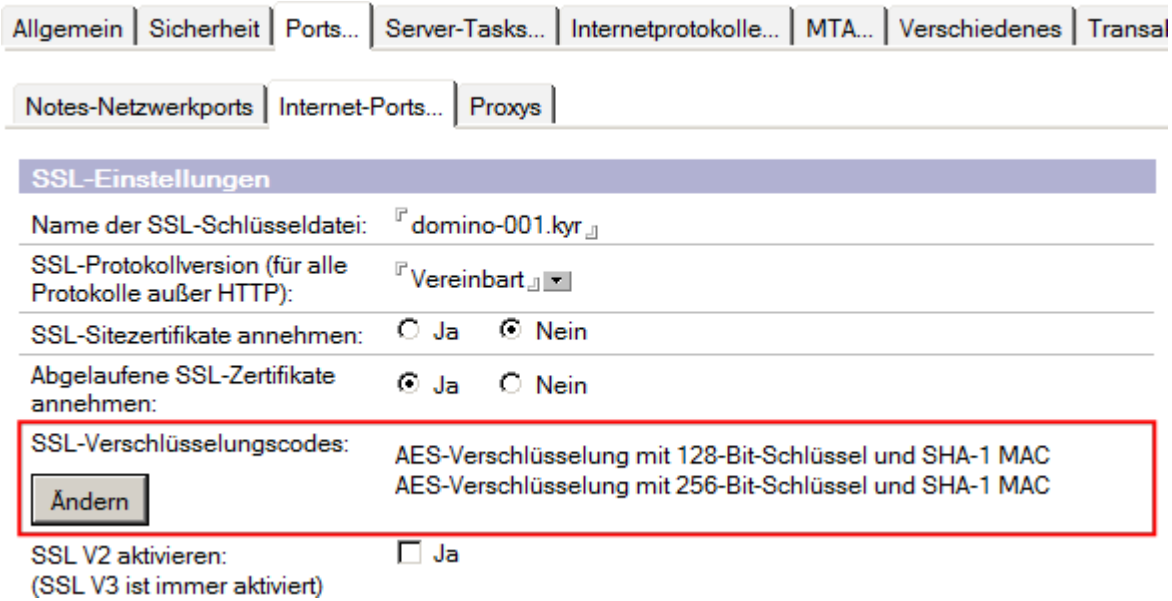
- „IBM Notes 9.0.1 FP3 IF3 and IBM Domino 9.0.1 FP3 IF2 provide support for Transport Layer Security version 1.2“
- HTTP, SMTP, LDAP, POP3 & IMAP.
- Perfect Forward Secrecy (PFS) via Ephemeral Diffie-Hellman (DHE) cipher specs for SSL/TLS
- New notes.ini SSL_DISABLE_TLS_10
- Implement Http Strict Transport Security (HSTS).
- IBM Notes and Domino Interim Fixes to support TLS 1.2, http://www-10.lotus.com/ldd/dominowiki.nsf/dx/TLS_1.2
- „TLS 1.2 will not be available for 8.5.x releases since the TLS 1.2 specification requires updated cryptographic libraries that are available only in Domino 9.0 […]“

Domino-Server lernt Neues: HSTS

- „The HTTP Strict Transport Security (HSTS) HTTP response header can be used by web servers to indicate that web clients should only communicate with them over HTTPS and never over HTTP.“
- HTTP_HSTS_MAX_AGE: Maximales Alter in Sekunden, Vorgabe ist 604.800 s = 1 Woche (bzw. 0, wenn HTTP-Port aktiv)
- HTTP_HSTS_INCLUDE_SUBDOMAINS=1: HSTS-Einstellung gilt auch für Unterdomänen, Vorgabe ist „deaktiviert“.
- HTTP_ENABLE_HSTS=0: HSTS deaktivieren, den Strict-Transport-Security-Header nicht senden.
- HTTP Strict Transport Security (HSTS), www.lotus.com/ldd/dominowiki.nsf/dx/HSTS
- geht auch mit Web-Seiten-Regel (ältere Domino-Versionen)

HTTPS mit Domino: Algorithmen einstellen

- bei SSL-Verbindungen einigen sich Web-Server und Browser bm Verbindungsaufbau auf ein Verschlüsselungsverfahren (Cipher)
- beide Seiten haben Listen mit Algorithmen und Schlüssellängen, die sie unterstützen, wobei weiter oben die stehen, die bevorzugt werden sollen
- Domino-Server: Einstellung im Server-Dokument:



HTTPS mit Domino: Algorithmen einstellen (forts.)

- bzw. Internet-Site-Dokument
  - Reiter “Sicherheit” →
  - Abschnitt “SSL-Sicherheit” →
  - Feld “SSL-Verschlüsselungscodes”
- neuere, sichere Verfahren fehlen?!
- keine Schablonenänderungen bei „kleinen“ Releases…
- komplette Einstellungen über notes.ini: SSLCipherSpec
- „überschreibt“ Einstellungen im Domino-Verzeichnis
- SSLCipherSpec (notes.ini-Einstellung), www.lotus.com/ldd/dominowiki.nsf/dx/SSLCipherSpec

Exkurs: SSLCipherSpec

- 2-/4-stellige Codes für Algorithmen und Schlüssellängen:
  - 03 - RSA_EXPORT_WITH_RC4_40_MD5
  - 04 - RSA_WITH_RC4_128_MD5
  - 05 - RSA_WITH_RC4_128_SHA
  - 06 - RSA_EXPORT_WITH_RC2_CBC_40_MD5
  - 09 - RSA_WITH_DES_CBC_SHA
  - 0A - RSA_WITH_3DES_EDE_CBC_SHA
  - 2F - RSA_WITH_AES_128_CBC_SHA
  - 33 - DHE_RSA_WITH_AES_128_CBC_SHA
  - 35 - RSA_WITH_AES_256_CBC_SHA
  - 39 - DHE_RSA_WITH_AES_256_CBC_SHA
  - 3C - RSA_WITH_AES_128_CBC_SHA256
  - 3D - RSA_WITH_AES_256_CBC_SHA256
  - 67 - DHE_RSA_WITH_AES_128_CBC_SHA256
  - 6B - DHE_RSA_WITH_AES_256_CBC_SHA256
  - 9C - RSA_WITH_AES_128_GCM_SHA256
  - 9D - RSA_WITH_AES_256_GCM_SHA384
  - 9E - DHE_RSA_WITH_AES_128_GCM_SHA256
  - 9F - DHE_RSA_WITH_AES_256_GCM_SHA384
- Beispiel: SSLCipherSpec = 9D9C3D353C2F0A
- Prioritätsreihenfolge durch Domino-Server festgelegt!

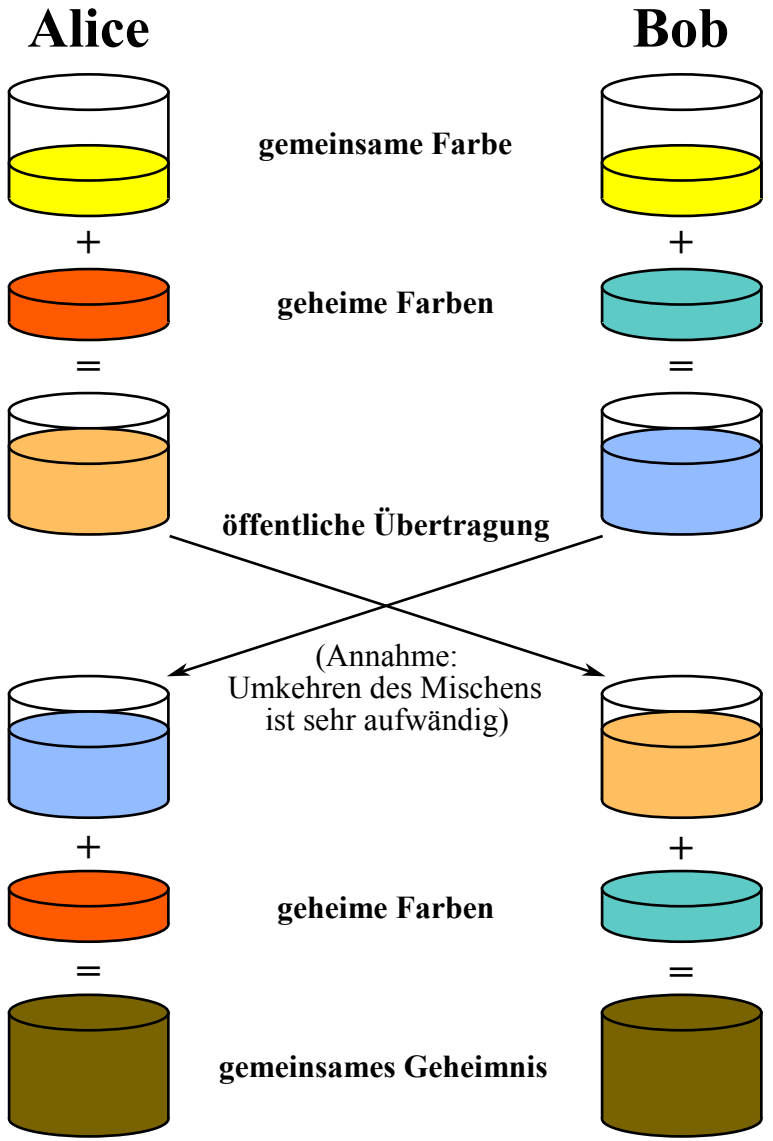
Exkurs: SSLCipherSpec (forts.)

- Mit Domino 9.0.1 FP3 IF2 kam TLS 1.2 - seitdem Einstellungen **ausschließlich** mit SSLCipherSpec!

- Sichere Vorgabe-Konfiguration:
  - Mit TLS 1.2:
    - 1. RSA_WITH_AES_256_GCM_SHA384
    - 2. RSA_WITH_AES_128_GCM_SHA256
    - 3. RSA_WITH_AES_256_CBC_SHA256
    - 4. RSA_WITH_AES_256_CBC_SHA
    - 5. RSA_WITH_AES_128_CBC_SHA256
    - 6. RSA_WITH_AES_128_CBC_SHA
    - 7. RSA_WITH_3DES_EDE_CBC_SHA
  - Mit TLS 1.0 oder SSLv3:
    - 1. RSA_WITH_AES_256_CBC_SHA
    - 2. RSA_WITH_AES_128_CBC_SHA
    - 3. RSA_WITH_3DES_EDE_CBC_SHA

Domino-Server lernt Neues: DHE

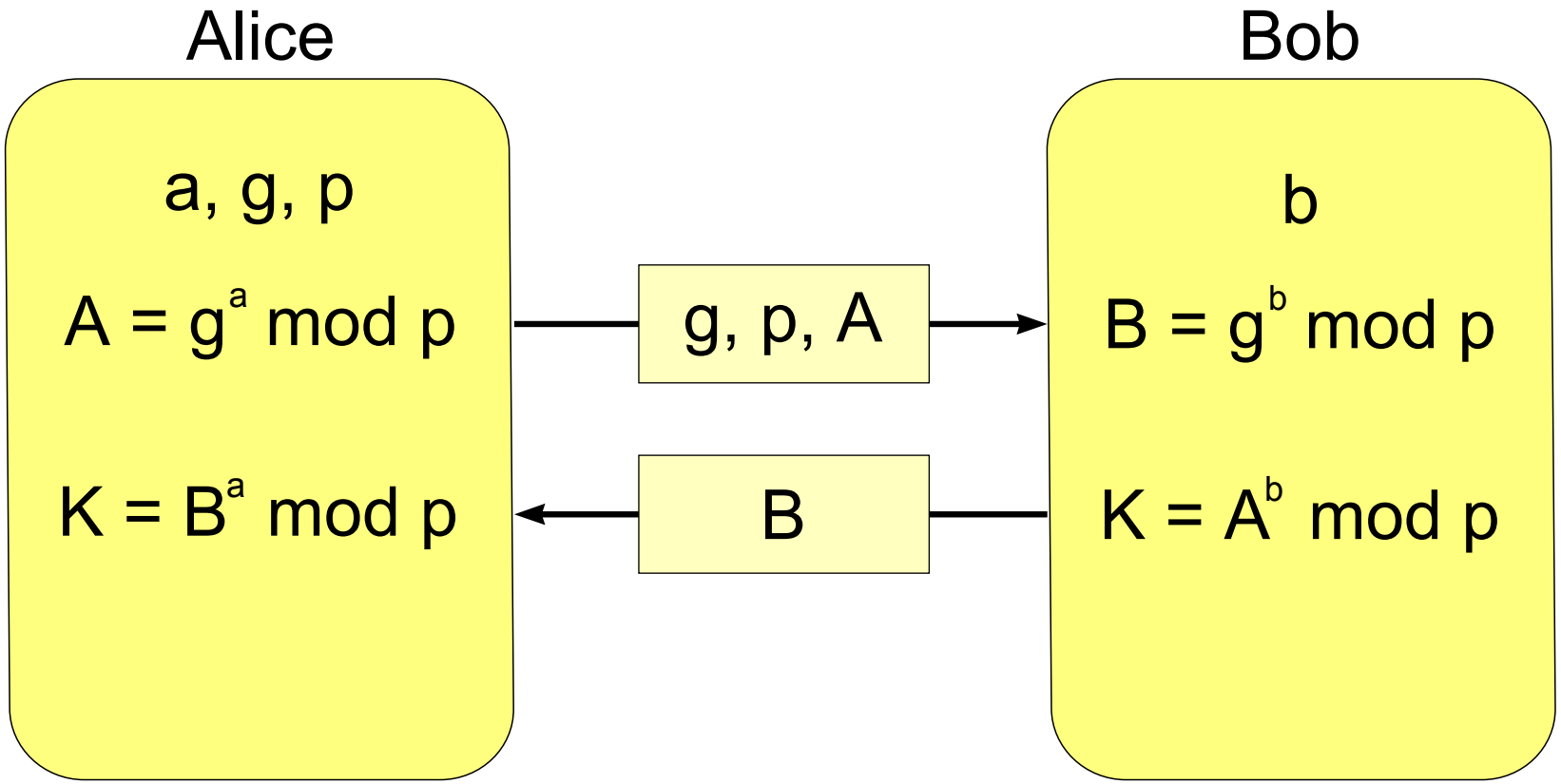
- Diffie-Hellman Key Exchange:
  Zwei Seiten einigen sich auf gemeinsamen Schlüssel, ohne ihn zu übertragen
- Forward Secrecy:
  Bei der Schlüsselaushandlung erzeuge „Einmalschlüssel", damit später nicht alle aufgezeichneten Sessions entschlüsselt werden können, wenn ein „Langzeit-Schlüssel" bekannt wird



https://commons.wikimedia.org/wiki/File:Diffie-Hellman_Key_Exchange_(de).svg

Diffie-Hellmann-Schlüsselaustausch

- etwas genauer:



$K = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p$

https://de.wikipedia.org/wiki/Datei:Diffie-Hellman-Schlüsselaustausch.svg

Domino-Server lernt Neues: DHE (forts.)

- „The DHE ciphers use Ephemeral Diffie-Hellman to provide Perfect Forward Secrecy (PFS), which protect against an attacker capable of passively recording all of the network traffic flowing into a server from later acquiring the server's private key and decrypting all of that recorded traffic. These ciphers significantly increase the security of your SSL/TLS traffic, at the cost of a potentially significant performance impact.“

- TLS Cipher Configuration, www.lotus.com/ldd/dominowiki.nsf/dx/TLS_Cipher_Configuration

Domino-Server lernt Neues: DHE (forts.)

- Neue Werte für SSLCipherSpec:
  - DHE_RSA_WITH_AES_256_GCM_SHA384 (9F)
  - DHE_RSA_WITH_AES_128_GCM_SHA256 (9E)
  - DHE_RSA_WITH_AES_256_CBC_SHA256 (6B)
  - DHE_RSA_WITH_AES_256_CBC_SHA (39)
  - DHE_RSA_WITH_AES_128_CBC_SHA256 (67)
- nicht vorgabemäßig aktiv, da DHE relativ performancehungrig

Domino-Server lernt Neues: ECDHE

- Elliptic Curve Diffie-Hellman Exchange (ECDHE)
- effizienter als DHE, daher besser für mobile Geräte
- dringend erforderlich bei iOS 9 und Mac OS X 10.11
  - erhöhte Anforderungen nach Apple App Transport Security, https://developer.apple.com/library/prerelease/ios/technotes/App-Transport-Security-Technote/
  - Vorgabe, wenn Apps für iOS 9 neu kompiliert werden
  - Ausnahmen können und müssen aktiv vom App-Entwickler eingestellt werden
  - keine selbstsignierten SSL-Zertifikate mehr

Exkurs: Apple App Transport Security

- „The server must support at least Transport Layer Security (TLS) protocol version 1.2.
- Connection ciphers are limited to those that provide forward secrecy [...]“
  - TLS_**ECDHE**_ECDSA_WITH_AES_* und TLS_**ECDHE**_RSA_WITH_AES_*
- „Certificates must be signed using a SHA256 or better signature hash algorithm, with either a 2048 bit or greater RSA key or a 256 bit or greater Elliptic-Curve (ECC) key.
- Invalid certificates result in a hard failure and no connection.“

Domino-Server lernt Neues: ECDHE (forts.)

- IBM Technotes:
  - Apple's App Transport Security prevents apps from connecting to a Domino server, www.ibm.com/support/docview.wss?uid=swg21966059
  - Q&A about IBM Traveler and Verse support for Apple iOS 9.x, www.ibm.com/support/docview.wss?uid=swg21962180
- IBM Notes Traveler betroffen?
  - "Apple iOS 9 is supported by IBM Traveler server 9.0.1.7"
- IBM Verse App
  - "Apple iOS 9 is supported by IBM Verse 9.1.2"

Domino-Server lernt Neues: ECDHE (forts.)

- ECDHE kam in 9.0.1 FP5 und Interims Fix für 9.0.1 FP4
  www.lotus.com/ldd/dominowiki.nsf/dx/TLS_Cipher_Configuration
- automatisch priorisiert: sicherer und gegenüber DHE effizienter
- Für 8.5.3 wird es keine Lösung mehr geben:
  "Elliptic Curve support will not be available for Domino 8.5.x releases since the specification requires updated cryptographic libraries that are available only in Domino 9.0 and above."

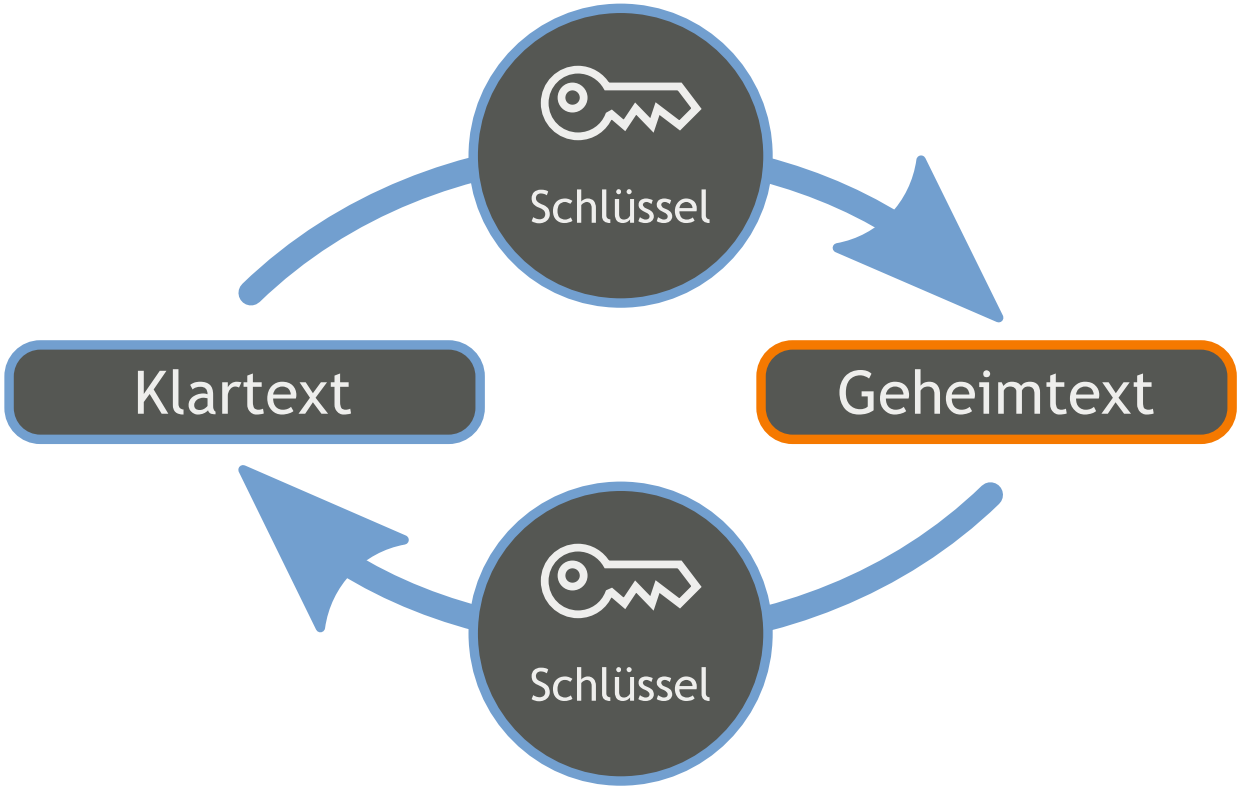
Pragmatische Einführung in die Kryptographie

Symmetrische Verschlüsselung

- symmetrisch: gleicher Schlüssel zum Ver- und Entschlüsseln
- 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“

Symmetrische Verschlüsselung (forts.)

- Verschlüsselung und Entschlüsselung mit dem gleichen Schlüssel
- blau: geheime Informationen; orange: öffentliche Informationen



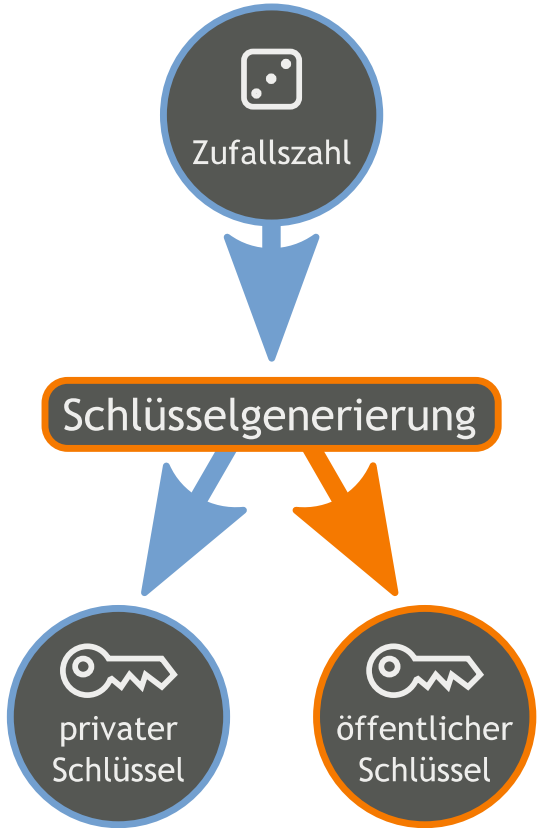
https://de.wikipedia.org/wiki/Datei:Orange_blue_symmetric_cryptography_de.svg

Asymmetrische Verschlüsselung

- asymmetrisch: zwei verschiedene Schlüssel zum Ver- und Entschlüsseln
- 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.)

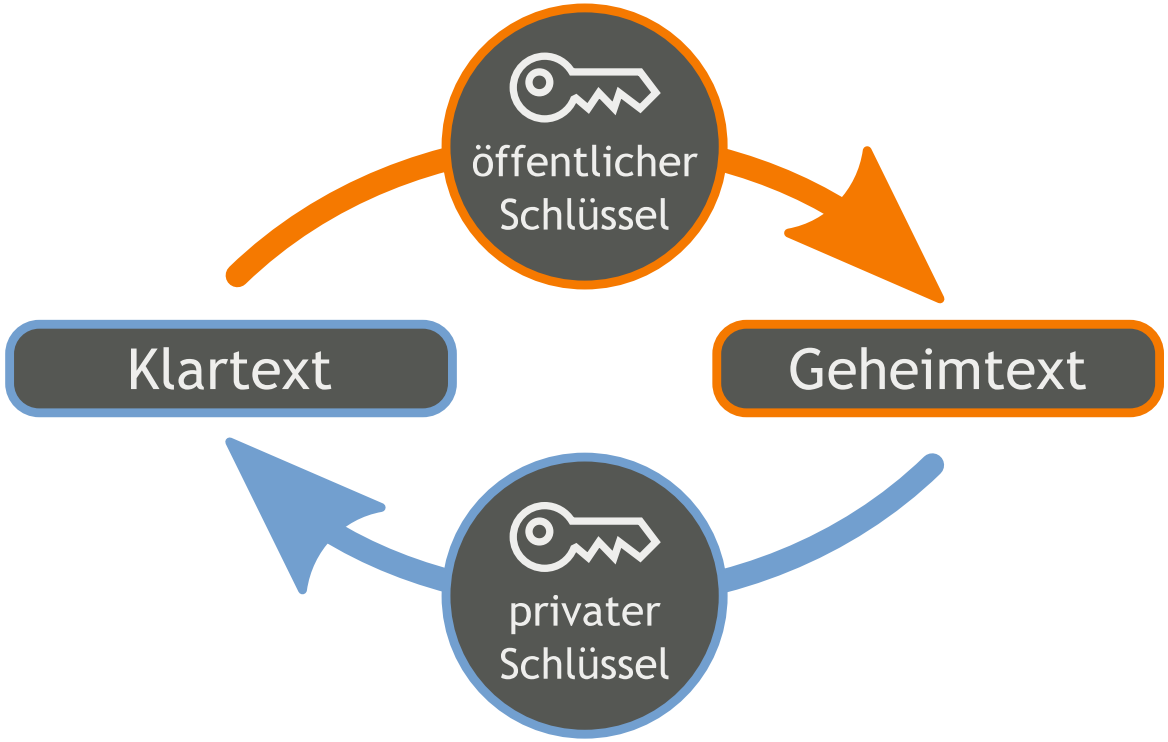
- Schlüsselpaargenerierung:
  Eingabe ist eine große Zufallszahl, Ausgabe das Schlüsselpaar



https://de.wikipedia.org/wiki/Datei:Orange_blue_public_private_keygeneration_de.svg

Asymmetrische Verschlüsselung (forts.)

- Verschlüsselung (öffentlicher Schlüssel) und Entschlüsselung (privater Schlüssel)
- orange: öffentliche Informationen; blau: geheime Informationen



https://de.wikipedia.org/wiki/Datei:Orange_blue_public_key_cryptography_de.svg

Asymmetrische Verschlüsselung (forts.)

- „Falltür"-Funktion:
  - in die eine Richtung geht es einfach…
  - aber man kann mit dem verschlüsselten Text (Chiffrat) und dem öffentlichen Schlüssel nicht auf die ursprüngliche Information zurück kommen.

- Asymmetrische Verfahren sind auch ein bisschen symmetrisch: man kann mit privatem Schlüssel verschlüsseln und dann mit öffentlichen Schlüssel entschlüsseln

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:

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
- (kryptografische) 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)

Signaturen (forts.)

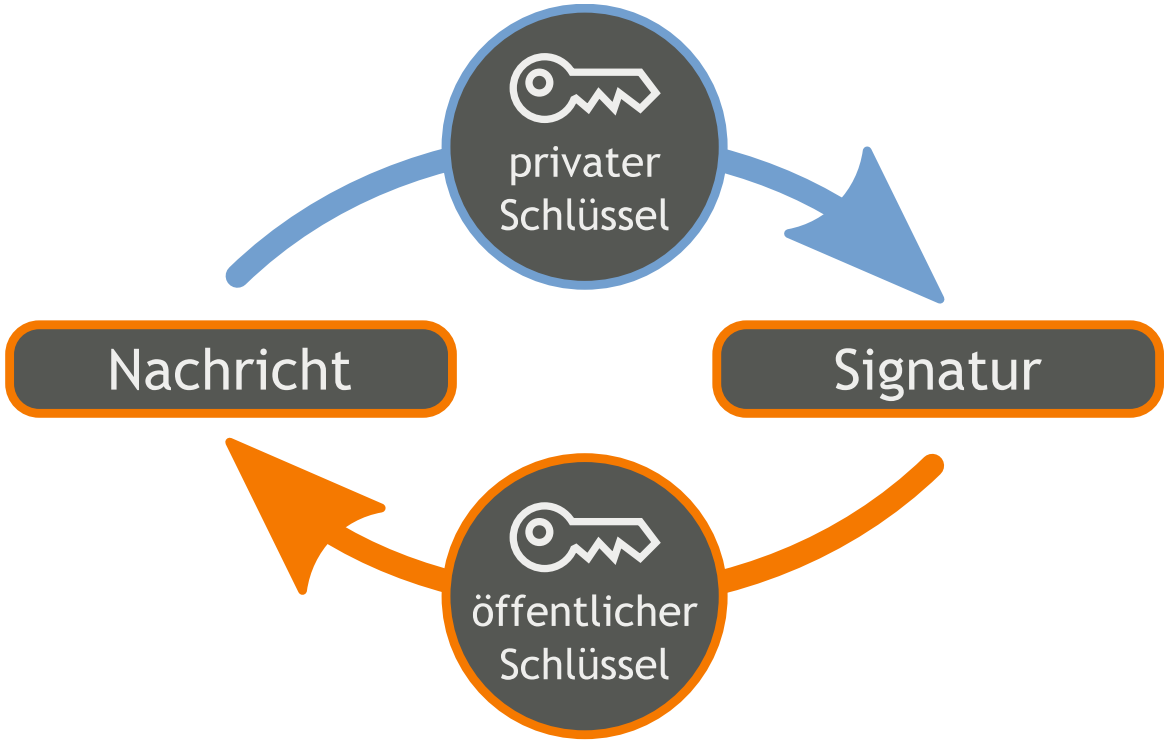
- Prüfen: verschlüsselten Hash-Wert mit öffentlichen Schlüssel entschlüsseln und mit selbst berechneten Hash-Wert vergleichen

- 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.

Digitale Signatur (forts.)

- Signieren mit dem privaten Schlüssel und Verifikation mit dem öffentlichen Schlüssel
- orange: öffentliche Informationen; blau: geheime Informationen



https://de.wikipedia.org/wiki/Datei:Orange_blue_digital_signature_de.svg

Authentifizierung

Authentifizierung vs. Autorisierung

- Authentifizierung: Wer bin ich?
- Autorisierung: Was darf ich?

Anmeldeprozess

- Anmeldung des Notes-Clients beim Domino-Server
- Ist der Benutzer wirklich der, der er behauptet zu sein?
- Ist der Server wirklich der, der er behauptet zu sein?
- gilt auch zwischen zwei Servern

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 anders herum

Passwort-Management

Sicherheitseinstellungen (Richtlinien)

- Kennwortverwaltung – Allgemein – Kennwortverwaltungsoptionen
- Benutzerdefinierte Kennwortrichtlinie für Notes-Clients verwenden: detaillierte Vorgaben für das Kennwort
- Kennwort überprüfen anhand der Notes-ID-Datei: Ja
  - Problem: ID gestohlen oder kompromittiert
  - Passwort nur auf ID-Datei – Passwortänderung wirkt nicht auf alte ID-Datei
  - Überprüfung anhand Digest in Personen-Dokument
- Internetkennwort bei Änderung des Notes-Client-Kennworts aktualisieren: Ja
- Andere Notes-basierte Programme fragen kein Kennwort ab (vermindert Sicherheit): Nein

Sicherheitseinstellungen (Richtlinien)

- Kennwortverwaltung – Allgemein
- Einmalige Anmeldung von Windows für Standard-Notes-Client aktivieren:
  - “SPNEGO”, gleich mehr
- Sperreinstellungen für Internetkennwörter
- Sperreinstellungen für Internetkennwörter des Servers überschreiben?
- Anmerkung: Der Server muss die Sperre für Internetkennwörter erzwingen, damit diese Richtlinieneinstellungen wirksam werden.

Gemeinsame Notes-Anmeldung

- Wenn die gemeinsame Notes-Anmeldung aktiviert ist, haben Notes-IDs keine Notes-Kennwörter mehr. Stattdessen wird zum Schutz der ID ein komplexes "Secret" verwendet. Dieses "Secret" wird anhand eines Microsoft Windows-Sicherheitsmechanismus verschlüsselt und auf den Computern der Benutzer lokal gespeichert.

Integrated Windows Authentication (IWA)

- für Eclipse-basierte Clients

- Integrated Windows Authentication (IWA) steht für bereitgestellte und Eclipse-basierte Clientanwendungen von Drittanbietern zur Verfügung und ermöglicht die SPNEGO-Authentifizierung für Eclipse-basierte Funktionen und Anwendungen im Notes-Client. Dies schließt Widgets und Live Text, Feeds, IBM Connections und Verbundanwendungen sowie integriertes IBM Sametime und Symphony ein. IWA funktioniert auch mit Produkten, die auf Eclipse basieren, aber nicht in Notes integriert sind, beispielsweise IBM WebSphere Portal mit SiteMinder und eigenständigem Connections 3.0 mit SiteMinder.

- Anmerkung: IWA kann nicht als Mechanismus zur Authentifizierung beim Start des Notes-Clients verwendet werden.

Föderierte Anmeldung

- Durch die Authentifizierung mittels föderierter Identität mit dem Security Assertion Markup Language-Standard (SAML) entfällt für Notes-Client-Benutzer die Eingabe eines Notes-Kennworts über die föderierte Notes-Anmeldung. Die Benutzer-IDs müssen in einer ID-Vault gespeichert werden, deren Domino-Server mit Hostnamen für Identitäts-Provider-Partnerschaften (IdP-Partnerschaften) konfiguriert ist. Der Inhalt der ID-Datei der Notes-Client-Benutzer wird nach dem Download aus der ID-Vault im Speicher auf dem Client abgelegt.

Sicherheit in Notes-Anwendungen

Echte Sicherheit vs. gefühlte Sicherheit

- „Echte Sicherheit“
  - Verschlüsselung!
  - Leser-/Bearbeiter-Felder
  - Zugriffskontrolle (ACL)
- „Gefühlte Sicherheit“
  - Hide-When-Formeln
  - Berechtigungen auf Ansichten, Selektionsformeln
  - versteckte Gestaltung
  - Selektive Replikation
  - Öffnen im Bearbeiten-Modus verhindern
  - Keine bearbeitbaren Felder in Maske

Datenbank-Eigenschaften

- Exportieren von Ansichtsdaten verhindern
- gilt auch für Kopieren als Tabelle
- verhindert nicht Drucken oder Kopieren von Dokumentinhalten

Ansichten-Eigenschaften

- Wer darf diese Ansicht verwenden?
- Verfügbar für Benutzer mit öffentlichem Zugriff

Masken-Eigenschaften

- Dokumente signieren, die diese Maske verwenden; $Signature; $SignatureStatus (geöffnetes Dokument)
- Standard-Verschlüsselungsschlüssel; SecretEncryptionKeys; $EncryptionStatus (geöffnetes Dokument)
- Standard-Lesezugriff für mit dieser Maske erstellte Dokumente; $Readers
- Wer kann mit dieser Maske Dokumente erstellen;
  aber: Kopieren und Einfügen, programmatisch per Agent erstellen, deshalb ACL!
- Drucken/Weiterleiten/Kopieren verhindern: $KeepPrivate = "1"
- Verfügbar für Benutzer mit öffentlichem Zugriff (Maske)
- $PublicAccess = "1" (Dokument)

Feld-Eigenschaften

- Verschlüsselung für dieses Feld aktivieren; $SealData
- Signieren beim Versenden oder Speichern im Abschnitt
- Mindestens Editorzugriff für Bearbeitung erforderlich
- Flags des Items: SIGN SEAL PROTECTED
- Eingabe-Formel

Kontrollierter Zugriff-Abschnitt

- Wer darf bearbeiten: Liste oder Formel
- Signieren des Abschnitt-Inhalts
- Item _unbenannt_: Editoren
- Item $Sig__unbenannt_: Signatur

Fragen?

jetzt stellen – oder später:

tbahn@assono.de
http://www.assono.de/blog
04307/900-401

assono
IT-Consulting & Solutions

Folien unter:
www.assono.de/blog/d6plinks/BC16-Weil-sicher-sicher-sicher-ist