assono

Cryptography in Lotus Notes/Domino – Pragmatic Introduction for Administrators

Belfast, 11-Nov-2010

Thomas Bahn

- graduated in mathematics, University of Hannover
- developing in Java and RDBMS since 1997
- dealing with Notes/Domino since 1999: development, administration, consulting and trainings
- frequent speaker at technical conferences about IBM Lotus Notes/Domino and author for THE VIEW

tbahn@assono.de
www.assono.de/blog
+49/4307/900-401

assono
IT-Consulting & Solutions

Agenda

- Modern Cryptography – The Basics
  - Encryption & Decryption
  - Hash Functions and Electronic Signatures
- Notes and Domino
  - Certificates and ID files
  - Encryption & Decryption
  - Electronic Signatures
  - Internet

Agenda

- **Modern Cryptography – The Basics**
  - **Encryption & Decryption**
  - Hash Functions and Electronic Signatures
- Notes and Domino
  - Certificates and ID files
  - Encryption & Decryption
  - Electronic Signatures
  - Internet

Modern Cryptography – The Basics

- **Cryptography** protects information by creating a **cipher text** from a **plain text**, thus only appointed persons can get to the protected information,
- where as steganography hides the information itself.
- Encryption is the process of transforming **plain text** into **cipher text**.
- Decryption is the process of transforming **cipher text** back into **plain text**.

Modern Cryptography – The Basics (cont.)

- First encryption techniques based on keeping the algorithms secret.
  - “Security by Obscurity”
  - inflexible
  - vulnerable
- Modern techniques nearly all use known algorithms with changing parameter values, called **keys**.
- A good encryption method is
  - publically known and available and
  - tested by many specialists for vulnerabilitys.

The Ultimate Encryption Method

- There is a mathematically proven 100% secure encryption method.
- It's easy to implement.
- It's absolutely fast.
- It's known for many, many years.
- It's called the **one-time pad**.
- And it's nearly useless...

One-Time Pad

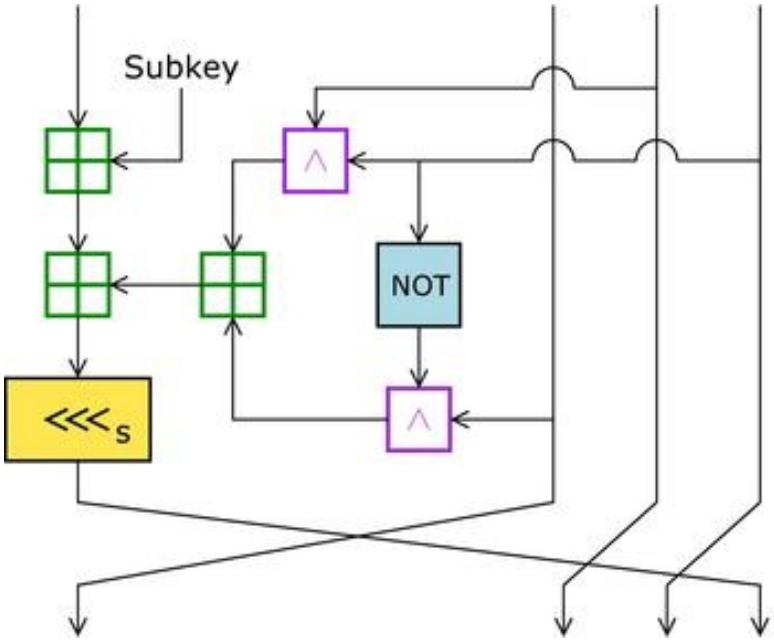
- based on a key of the **same length** as the plain text
- Key must be created **absolutely randomly**.
- Each key must be **used only once** (hence “one-time”).
- Key must be transferred to the reader.
- You replaced the problem of securely transporting the message by transporting the key...
- Only one advantage: You can create and distribute a list of keys in advance (e.g. in the form of a pad).
- But the list could be “found” in the meantime.

Symmetric Methods

- In **symmetric algorithms**, the same key is used for encryption and decryption.
- You have always the problem to securely transfer the key and keep it secret.
- You need to have a different key for each recipient.

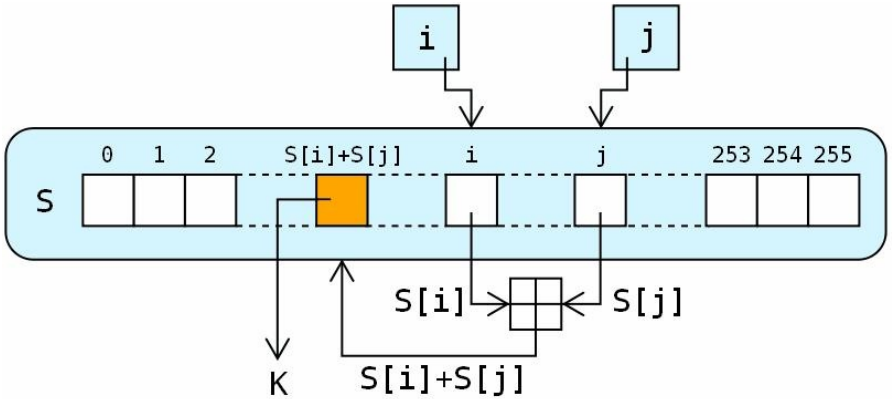
Symmetric Method: RC2

- 64 bit block cipher by Ronald Rivest, 1987
- RC: Rivest Cipher or Ron's Code
- created for Lotus
- "exportable" from US
- algorithm was kept secret
- published 1996 in the Usenet
- in Notes:
  - field encryption
  - encryption of ID files



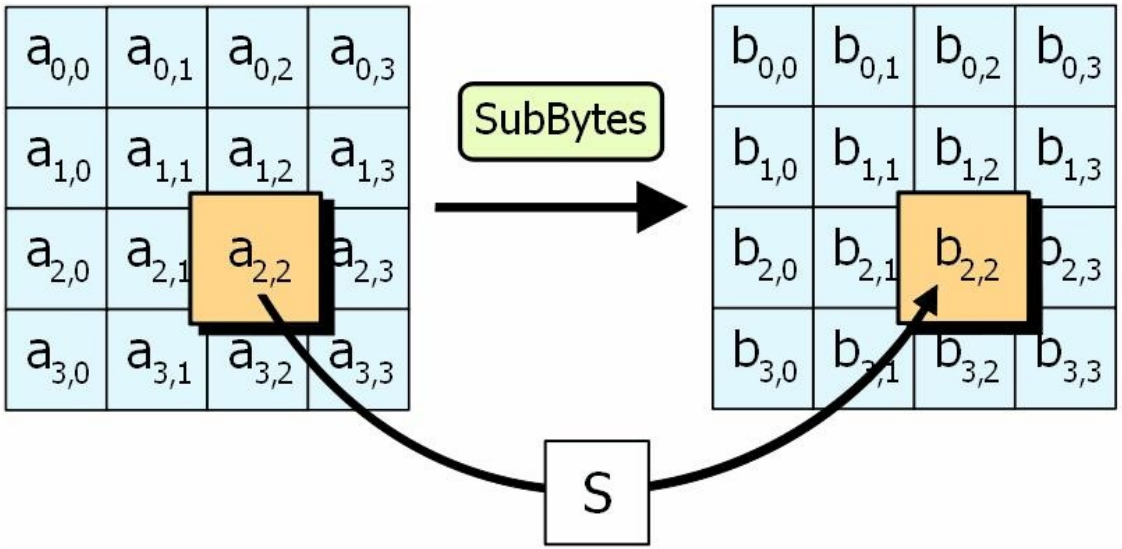
Symmetric Method: RC4

- stream cipher by Ronald Rivest, 1987
- variable key length (8 to 128 bit, normally 64 bit)
- Algorithm creates a “random” key of any length, which is used like in the one-time pad technique.
- It was kept secret, but published 1994 in the Usenet.
- simple to implement
- very fast
- weak for short messages
- in Notes:
  - network-encryption



Symmetric Method: AES

- Advanced Encryption Standard, October 2000
- Algorithm chosen as new encryption standard to succeed DES and Triple-DES.
- It was created by Vincent Rijmen and Joan Daemen: Rijndael algorithm.
- in Notes 8+:
  - encryption of ID files
  - SSL

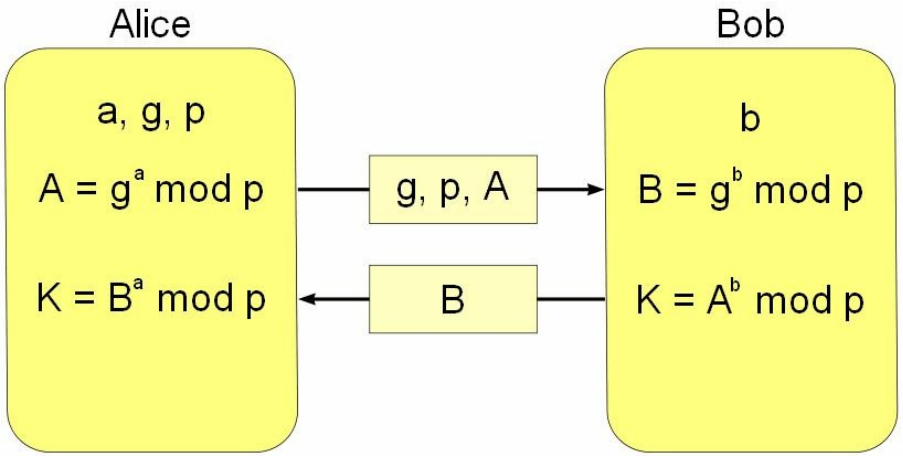


Mathematical Excursion

- **one-way function**: easy to calculate, hard to invert
- a^b mod n is such a one-way function.
- Inversion is called **discrete logarithm**.
- No efficient algorithm is known (yet) to calculate the discrete logarithm.
- Multiplication of (big) prime numbers is another one-way function, its inversion is called **factorisation**.
- **trapdoor function**: one-way function with a shortcut for the inversion (= decryption)

Key Distribution Problem

- Diffie-Hellman(-Merkle) key exchange
- discovered 1974 by Whitfield Diffie, Martin Hellman and Ralph Merkle
- Key itself is calculated on both sides, not transferred.
- Both sides keep a secret (a and b).
- K is used as key.
- This key and a symmetric algorithm is used for encryption.



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

RSA

- **RSA**, created 1977 by Ronald L. Rivest, Adi Shamir und Leonard Adleman
- most important and known **asymmetric algorithm**
- more flexible then DH(M), can by used for encryption
- It is based on the multiplication of big prime numbers,
- with a shortcut for decryption.
- in Notes:
  - ID files
  - encrypted emails

Asymmetric Methods

- **Asymmetric algorithms** use different keys for encryption and decryption.
- The key used to encrypt a message for you can be **public**, e.g. published in a directory, key server etc.
- The key used to decrypt must be kept **private**, thus nobody but you can decrypt messages intended for you.

Asymmetric Methods (cont.)

- **No more key transport problem!**
- And the secret private key is **only stored once** in your environment (no need to transport).
- The **same public key** can be used by all senders.
- Many asymmetric algorithms are symmetric in another way: Messages encrypted with the private key can only be decrypted with the public key.
- Only the owner has the private key, and if the cipher text can be decrypted with his public key, it must be him, who encrypted the message.

Symmetric vs. Asymmetric

- Symmetric algorithms are faster at same level of security,
- but have the key distribution problem.
- What to do?

Hybrid Algorithms

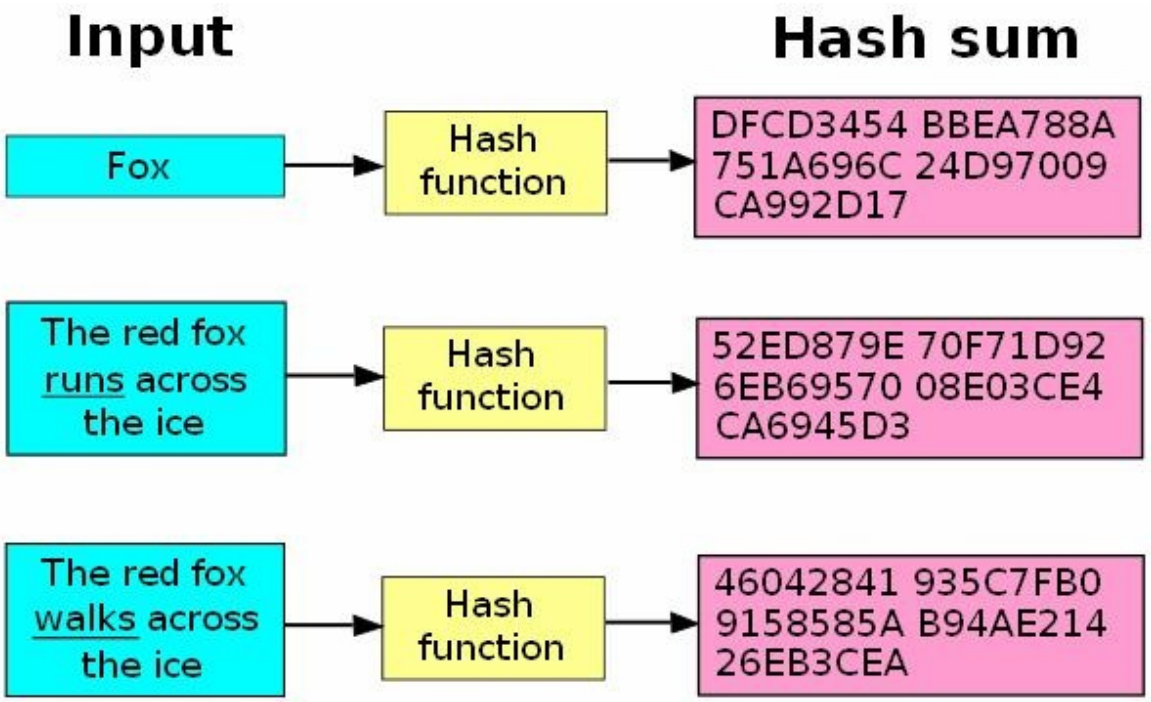
- Combine both methods:
  - A random key is created and distributed with DH(M) or RSA.
  - This random key is used to encrypt the message with a symmetric algorithm.
  - Only the appointed recipient can decrypt the key and with it the ciper text.
- For more recipients, you only have to encrypt the (short) random key multiple times, not the whole message.

Agenda

- Modern Cryptography – The Basics
  - Encryption & Decryption
  - **Hash Functions and Electronic Signatures**
- Notes and Domino
  - Certificates and ID files
  - Encryption & Decryption
  - Electronic Signatures
  - Internet

Hash Functions

- **Hash functions** return results with a constant length.
- example: modulo operation, @Password function



Hash Functions (cont.)

- Minimal changes to the input result in maximal changes of the output.
- **Cryptographic hash functions**:
  Changing one bit of the input results in an average change of the output of 50%.
- known algorithms: MD4, MD5, SHA-1
- in Notes:
  - SSL
  - signed documents and emails

Electronic Signatures

- also a hybrid technique
- The hash of the message is calculated, encrypted (e.g. with RSA) and send along with the message.

- Everybody can decode the encrypted hash value and calculate the hash of the message himself.
- If both values are identical, the message hasn't been changed and was created by the original sender.

It's All About Trust

- Digital data can easily be changed – without any traces.
- Electronic signatures can prove the authenticity and integrity, but the **public key must be genuine**.
- If somebody **you trust** had electronically signed the name (e.g. NotesName) of the other person together with his public key, this would prove its genuineness.
- The name, public key and this signature together are called **certificate**.
- Normally, certificates have only a limited lifetime and must be prolonged (=recertified) to remain valid.

It's All About Trust (cont.)

- The trusted entity creating certificates is called **certifier** or **certification authority** (**CA**).
- You can also easily have a complete hierarchy of CAs.
- This is called **public key infrastructure** (**PKI**).
- in Notes:
  - registration of new certifiers, servers and users
  - CA process
  - ID files
  - authentication (client-server or server-server)

Agenda

- Modern Cryptography – The Basics
  - Encryption & Decryption
  - Hash Functions and Electronic Signatures
- **Notes and Domino**
  - **Certificates and ID files**
  - Encryption & Decryption
  - Electronic Signatures
  - Internet

Certificates and ID files in Notes and Domino

- When you create a Domino infrastructure and configure your first server, a certifier ID is created.
- This is the certifier = Certification Authority for the new organisation and used to sign every other ID.
- Like any other ID file, it contains (among other things) the **NotesName**, current date, expiration date, the **public** and the **private key** along with the electronic signature of all this information, i.e. the **certificate**.
- The certified public key is also stored in the Domino Directory.
- OU certifier and their ID files are similar.

A Certifier Document in the Domino Directory

Certifier [NOTES CERTIFIER] :/TBahn

Basics | Recovery Configuration | Contact Information | Other | Administration

Basics

Certifier type:	Notes Certifier
Certifier name:	/TBahn
Issued by:	/TBahn
Issued to:	O=TBahn
Alternate names:	
Primary key identifier:	1WQBN P7W5S Y496B 9X1D8 CNAUG 944BC
International key identifier:	1G4MN GE34K 3YQ5W RAV18 D1G43 S7465
Current key strength:	Compatible with all releases (630 Bits)
Current key creation date:	04.03.2007 11:25:37
Certified public key:	03001D02 DF4FE4E3 07G01607 G002B701 5D585E03 G0030200 01208600 7EC53E00 947225G0 024FG002 83C53E00 93722500 83C53E00 40012600 01A07700 7EC53E00 947225G0 024FG002 83C53E00 93722500 83C53E00 40012600 4F3D5442 61686E4F 3D544261 686E4256 0400312E 30004243 01000342 41010030 424C0200 76024E4E 4F002F93 67AFA6CB 324C9CB3 4D2C3C3A

Registration of New Users And Servers

- When a new user or server is registered, a **key pair is created** and a **certificate issued** by the certifier or a OU certifier.
- This information is stored encrypted with the user's password **in the ID file** (only some information, as the NotesName is not encrypted).
- **In the person or server document** respectively the certified public key is stored.
- This is why you need access to the certifier's ID file – or use the CA process.
- In this case the certificates are stored in admin4.nsf instead of in the certlog.nsf.

Control Key Details When Registering New Users or...

Register Person -- New Entry

Basics
Mail
Address
ID Info
Groups
Roaming
Other

☑ Create a Notes ID for this person
☐ Use CA process

Certifier ID Information

Certificate expiration date:
10.11.2012 00:18:39

Certifier ID... /TBahn

Public key specification:
Compatible with 6.0 and later (1024 Bits)
Compatible with all releases (630 Bits)
Compatible with 6.0 and later (1024 Bits)
Compatible with 7.0 and later (2048 Bits)

License type:
North American

☑ In Domino directory
☐ In file: Set ID File... d:\Notes\Notes852\ids\people\user.id
☐ In mail file
☐ In personal address book
☐ In Notes ID vault

☑ Advanced New Person Migrate People... Import Text File...

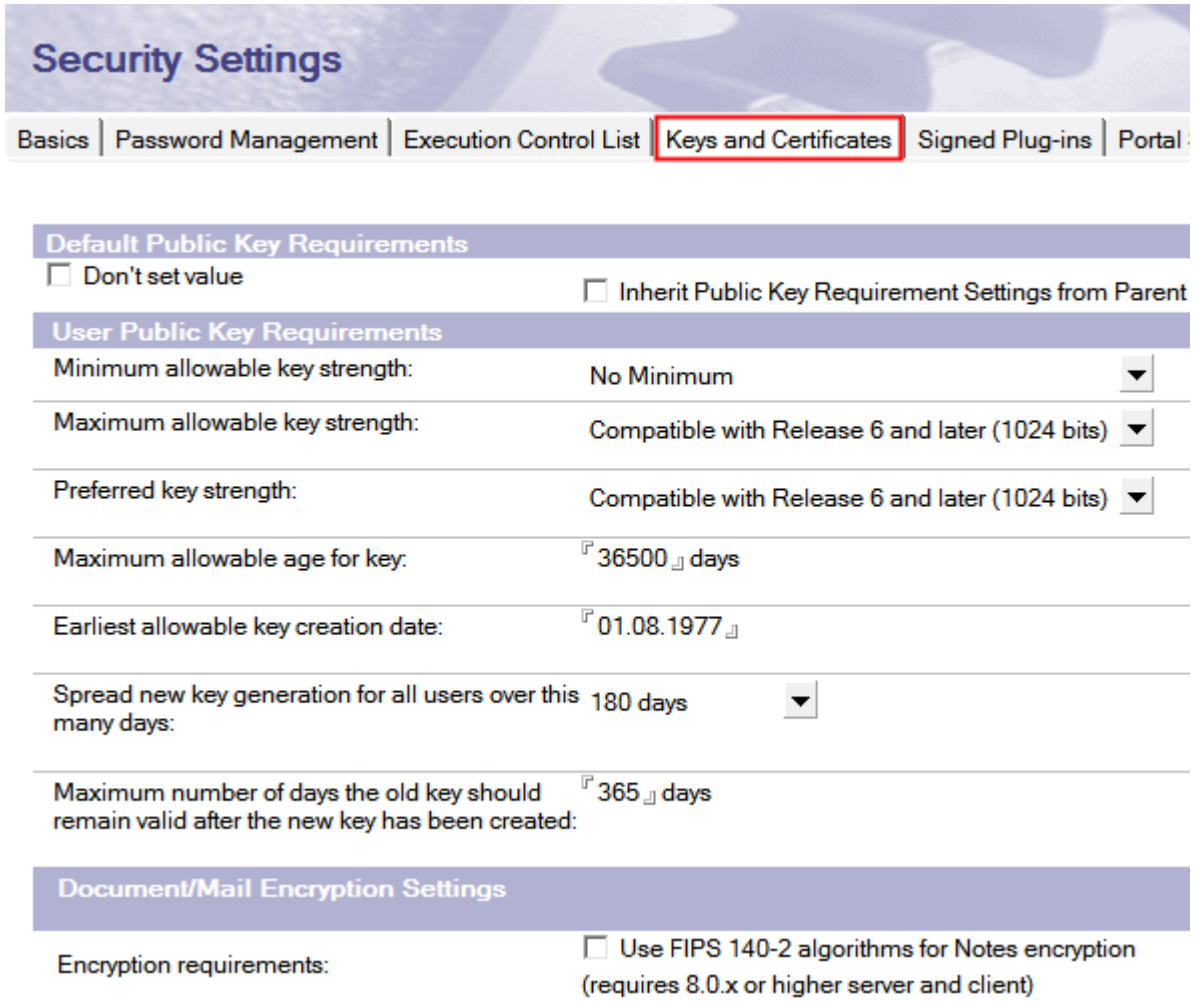
Registration Queue (local):

Benutzername	Registrierungsstatus	Datum

Register All Register Delete Options... Views... Done

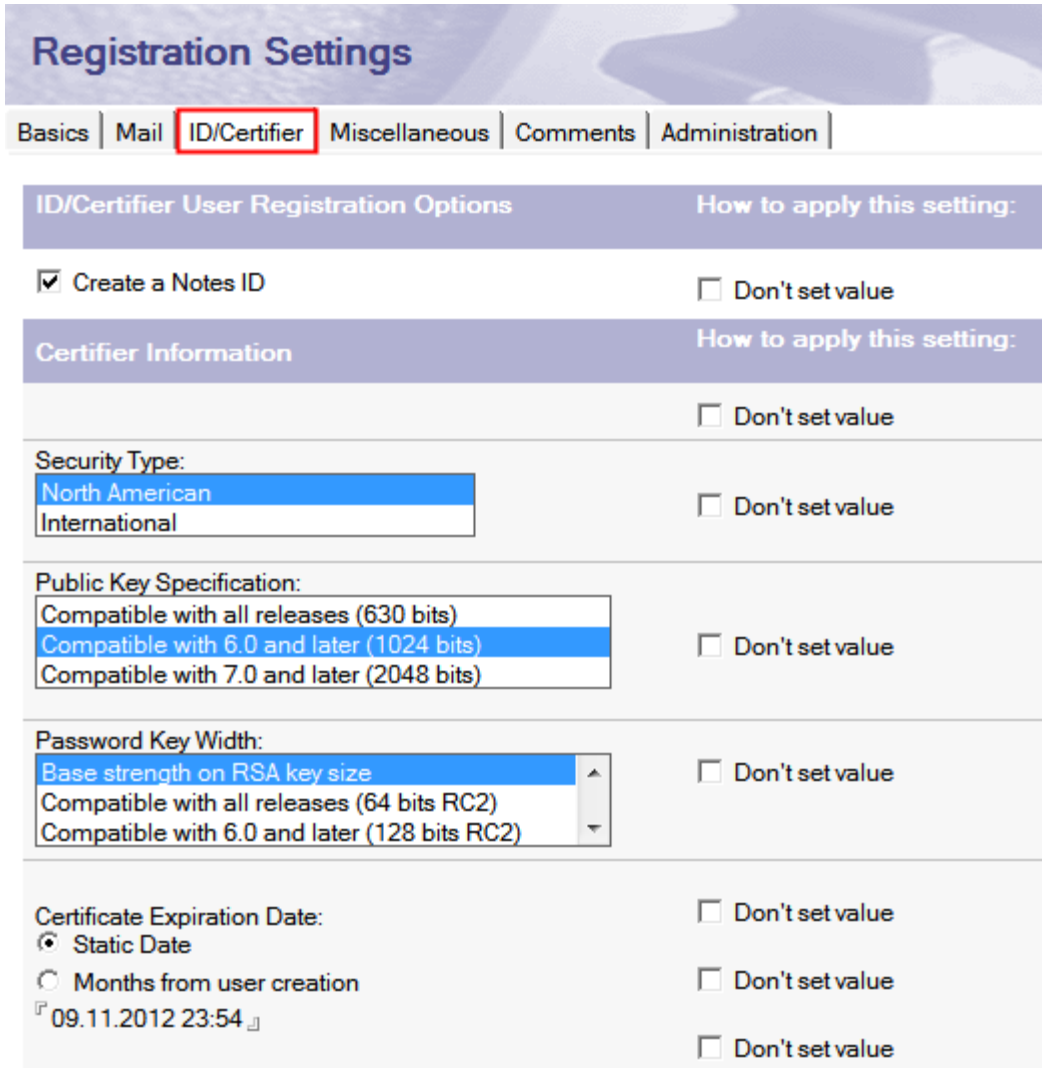
Use Security Settings And...

- With policies and security settings you can control the defaults (e.g. lifetime) and key strength:



Use Registration Settings

- With policies and registration settings you can control the even more details, like the public key length:



Person Document With a Notes Certificate

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

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

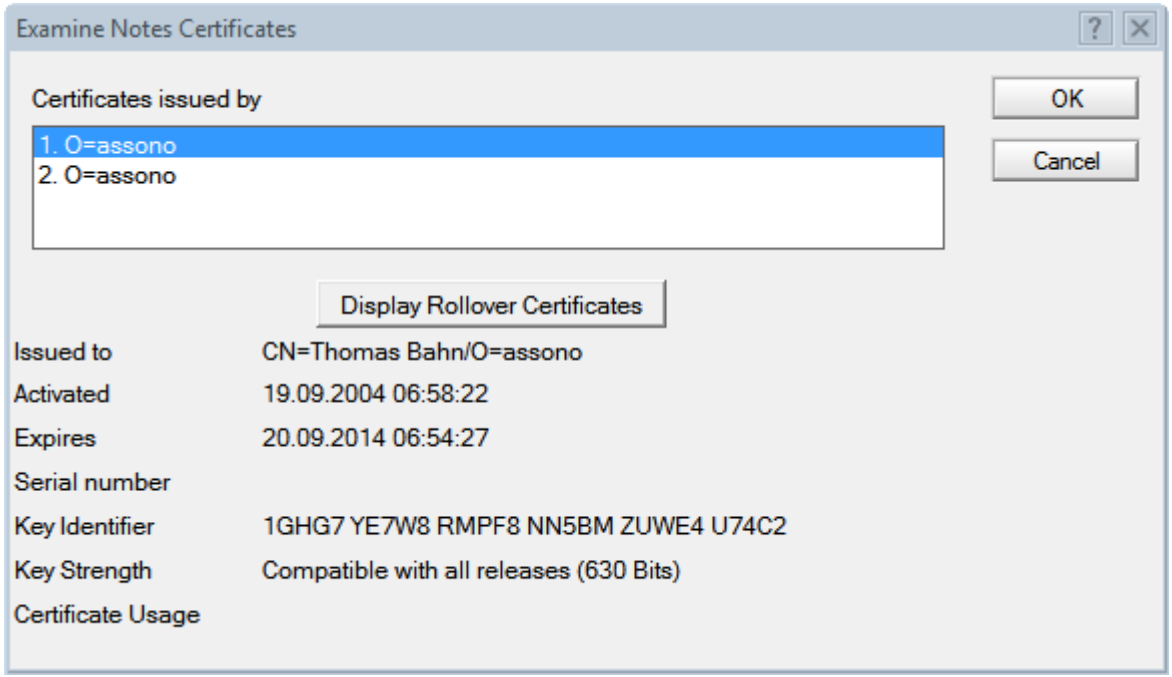
Notes Certificates | Internet Certificates | Flat Name Key

Notes Certificate(s)

Notes certificate:	Present
Primary key identifier:	1GHG7 YE7W8 RMPF8 NN5BM ZUWE4 U74C2
International key identifier:	16P5S MT822 5TUWV 47FZ8 HH7W3 Z24C4
Current key strength:	Compatible with all releases (630 Bits)
Current key creation date:	20.09.2004 06:58:22
Notes certified public key:	03002E02 A040F2C8 08G01617 G0020654 C5C27C03 G0030200 01208600 63CF2000 156F25G0 024FG002 63CF2000 146F2500 6C732000 597D25C1 01A07700 63CF2000 156F25G0 024FG002 63CF2000 146F2500 6C732000 597D25C1 4F3D6173 736F6E6F 434E3D54 686F6D61 73204261 686E2F4F 3D617373 6F6E6F42 56040031 2E300042 43010003 42410100 30424C02 0076024E 4E4F00B7 BEB419D7 18769C5F B192ED8C 87D71CDE 741E50F1 E8059919 D4C9F3A4 B4B8FB6A 29FD4BC1 F866201D 8DD3612B BC9913D8 7D8824AA 69B5E6CD 889CDA41 4E1DBE96 F9B3F828 68491C90 3E144F48 0635454E 03000100 014D4108 00F6F46B 67B780A6 147E0050 55525341 465632F8 0E0249D3 CAFADF54 C94ACECF 7F4E281C 4BD77F4F C7EE82F0 12340458 A6650F15 7EE8E9BF BAFB0AD1 E51672C9 A92EC12A E8A8B23F 227CA145 656DA330 B4CB0BE7 F93F7A67 9C528B01 AC4ED91C 42560400 312E3000 42430100 03424101 0030424C 02G00102 4E4E4000 B9F17FA0 7DBF60AE 390C8B4E F3563339 1BE28359 22AE9D8B BEB0F156 C524D434 A06C60A8 1BB79286 B00EA7A8 7EC67A4C 76E14EA6 016A5BAD E7CAA224 B4D890CF 454E0300 0100014D 41080017 E015E3EE 6C96076F 00505552 5341464D DB4AA9D3 EDCC4B3A 790DCEBD 4E6C92C4 0E25F62D FC6CCBB5 BF3B2114 662C586D 26ADAEF8 36A332A8 3FB7969B 32E5D4BD 2BF499EE 1A4D271F 43FADB77 6132A7FE 7D97E2C0 EA13D145 4C6841DA 9A0E

Check the Details of the Notes Certificates

Examine Notes Certificate(s)

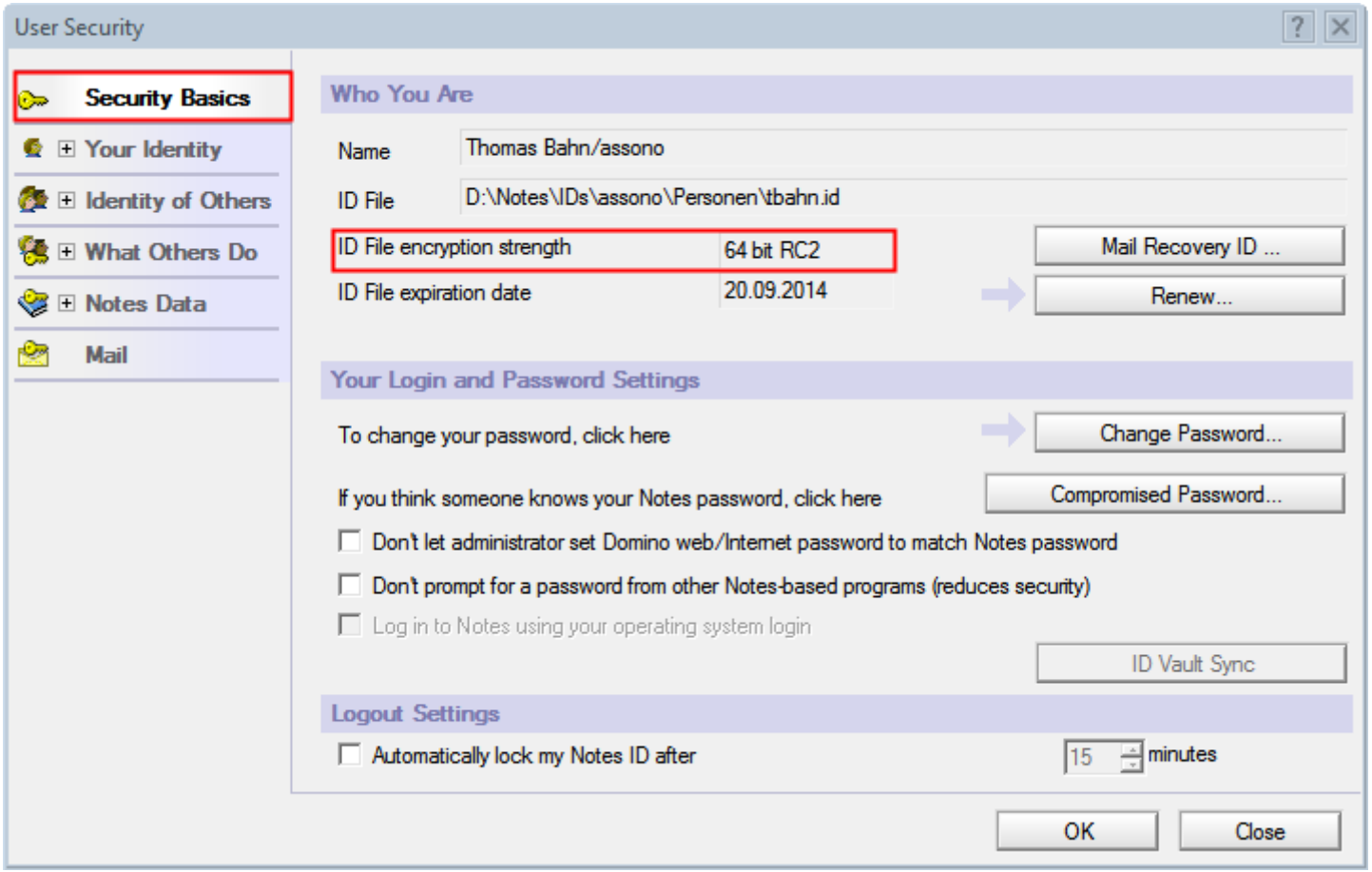


ID files

- ID files contain (among other information):
  - NotesName
  - private key
  - public key
  - certificate
  - certified public key of the certifier
  - internet certificates (optional)
  - secret keys (optional)
- Nearly all information are encrypted with the password entered at registration time (or to be more precise: encrypted with a key calculated from the entered password).

Details of Your ID File

- Open Security – User Security...



Details of Your ID File (cont.)

- Your Certificates

User Security

Security Basics
Your Identity
Your Names
Your Certificates
Your Smartcard
Identity of Others
What Others Do
Notes Data
Mail

Certificates in your ID file

Your certificates provide a secure way to identify you to Notes and other programs. Your ID may contain certificates used to secure Notes communications as well as certificates used with the Internet.

All Certificates — Includes your Internet and Notes certificates, and certificates for the certificate authorities that issued your certificates.

Type	Issued To	Issued By
	domino-001.assono.de	domino-001.assono.de
	tbahn@assono.de	domino-001.assono.de
	Thomas Bahn/assono	/assono
	Thomas Bahn/assono	/assono
	/assono	/assono

Get Certificates...
Other Actions...

Selected item

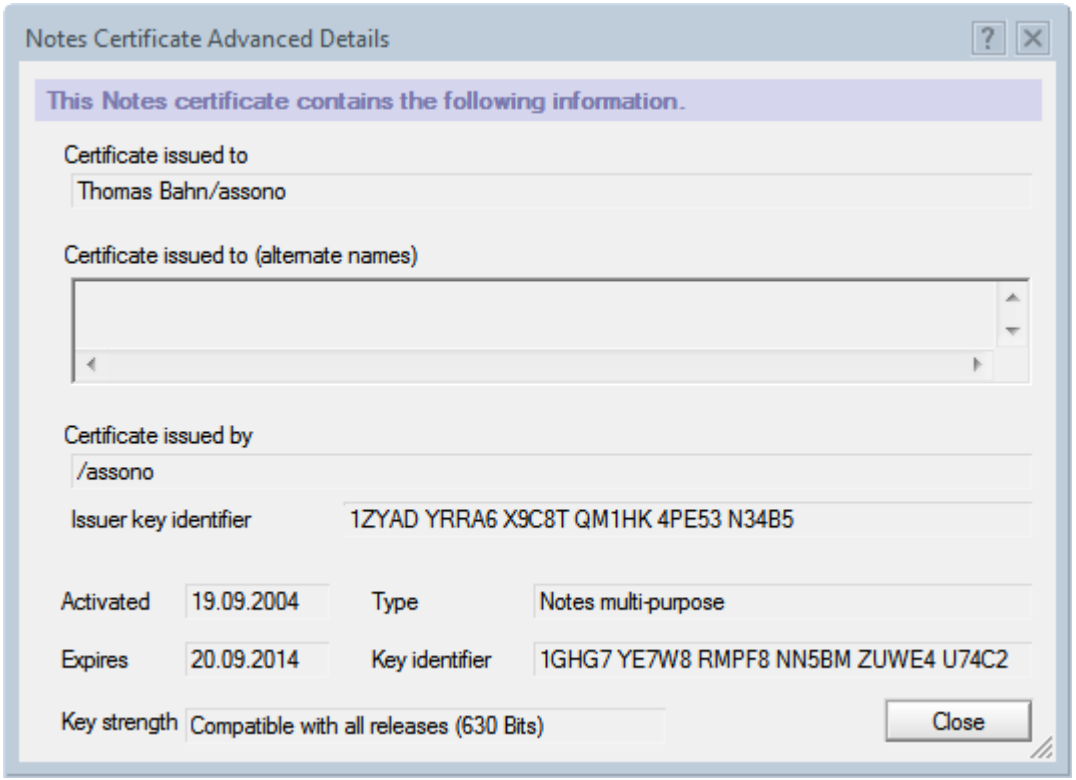
Issued to	domino-001.assono.de	(Email)	
Issued by	domino-001.assono.de	(Email)	
Activated	12.08.2008	Type	Internet certificate authority
Expires	13.08.2009	Fingerprint	C925 E00F 956B F5D2 62BC 205F 0E56 9D3A

Advanced Details...

OK Close

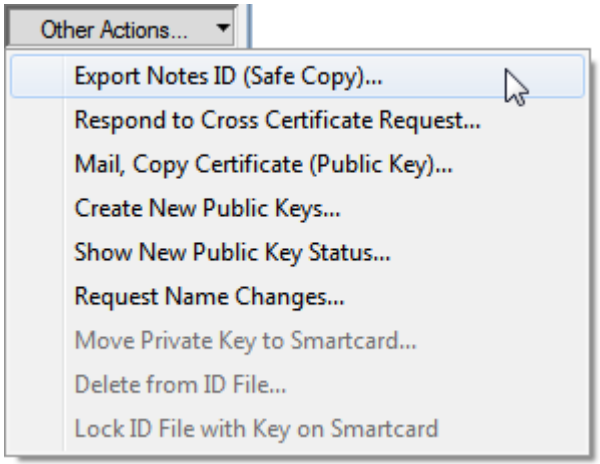
Details of Your ID File (cont.)

- Advanced Details...



Details of Your ID File (cont.)

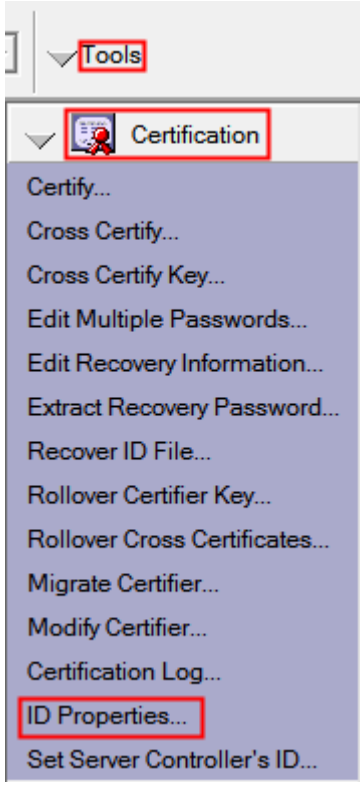
- Other Actions...



- A safe copy only contains unencrypted data like the NotesName, some dates and the certified public key.

To Investigate Other ID Files

- Use the tools on the configuration tab in the Admin client:



Notes Cross Certificates

- In a Domino organisation servers and users can authenticate using the certified public keys and the certification hierarchy.
- But how can you trust users and servers from other domains, not certified with a certifier you trust?
- Notes Cross Certificates are the answer!
- Just take a foreign ID file with NotesName, dates, public key etc. and create an electronic signature with a certifier you trust, i.e. your ID (user), a server ID, an OU or your domain's certifier.

Notes Cross Certificates (cont.)

- The cross certificate is stored in your personal address book (if signed with your user ID) or the public Domino Directory.
- To authenticate the foreign user or server, you check the cross certificate instead of the original certificate.

A Cross Certificate Document in the Domino Directory

Cross Certificate

Basics | Administration

Basics

Certificate type:	Notes Cross-Certificate
Issued By:	/TBahn
Issued To:	/assono
Alternate names:	
Combined Name:	O=TBahn:O=assono
Comment:	
Organizations:	O=TBahn:O=assono
X.509 certificate	Not Available
Primary key identifier:	1ZYAD YRRA6 X9C8T QM1HK 4PE53 N34B5
International key identifier:	1ZYAD YRRA6 X9C8T QM1HK 4PE53 N34B5
Current key strength:	Compatible with all releases (630 Bits)
Certified Public Key:	03003102 B3E40BAA 07G01608 G00E0120 8800EF77 G00202G0 034FG002 86860200 13732500 FC790200 568125C1 01A08800 EF77G002 02G0034F G0028686 02001373 2500FC79 02005681 25C14F3D 54426168 6E4F3D61 73736F6E 6F425604 00312E30 00424301 00034241 01003042 4C020076 024E4E50 00358984 A6DB35B4 8F56EB09

ID Recovery

- You can **store recovery information in the ID files**, which can be used to decrypt the private data.
- The recovery information is stored encrypted and in a way, you can control, who and how many persons together can decrypt it.
- Backups of the ID files are send to a mail(-in) db. These are used in the case, you have to create a new ID for the user.
- For this to work, you have to prepare the certifier used to register new users. Existing users must be recertified.
- or use the ID Vault (if on 8.5)

Authentication Process

- When a user logs into a server, two checks are performed.
- **Validation of the public key**: the client sends the NotesName of the user and his public key from his ID file to the server, which can compare this with the one stored in the Domino Directory.

Authentication Process (cont.)

- **Mutual authentication** using a challenge/response process:
  - The server creates random number,
  - encrypts it with the user's public key and
  - sends the result to the client.
  - The client decrypts the number,
  - encrypts it with the servers public key and
  - sends it back to the server.
  - The server decrypts it again and
  - compares it to the original number.

Authentication Process (cont.)

- If both numbers are identical, the client must have access to the user's private key, which only the user can have and open with his password.
- Then the process is repeated with changed roles.

Servers Might Have Requirements For Public Keys

Server: **Notebook-016-8.5.2/TBahn** notebook-016-86

MTAs... | Miscellaneous | Transactional Logging | Shared Mail | DAOS | Lotus Traveler | Administration

Administration

Owner:	
Administrators:	

Public Key Requirements

Minimum allowable key strength:	No Minimum
Maximum allowable key strength:	Compatible with Release 6 and later (1024 bits)
Preferred key strength:	Compatible with Release 6 and later (1024 bits)
Maximum allowable age for key:	36500 days
Earliest allowable key creation date:	01.08.77
Don't automatically generate a new key before:	05.10.2110
Maximum number of days the old key should remain valid after the new key has been created:	365 days

Notes Certificate(s)

Notes certificate:	Present
Primary key identifier:	1NDQ5 KN5QE D8Q44 Y4FP9 YQ4XV F74BG
International key identifier:	1NDQ5 KN5QE D8Q44 Y4FP9 YQ4XV F74BG
Current key strength:	Compatible with 6.0 and later (1024 Bits)
Current key creation date:	05.10.2010 22:01:37
Certified public key:	03007A02 4B73C31C 07G0161D G00283EE 512F3203 G0030200 0120A200 2A817300 B37725G0 024FG002 2C817300 B2772500 4C597300 5F0626C1 01A0A200 2A817300 B37725G0 024FG002 2C817300 B2772500

Agenda

- Modern Cryptography – The Basics
  - Encryption & Decryption
  - Hash Functions and Electronic Signatures
- Notes and Domino
  - Certificates and ID files
  - **Encryption & Decryption**
  - Electronic Signatures
  - Internet

Encryption and Decryption in Notes and Domino

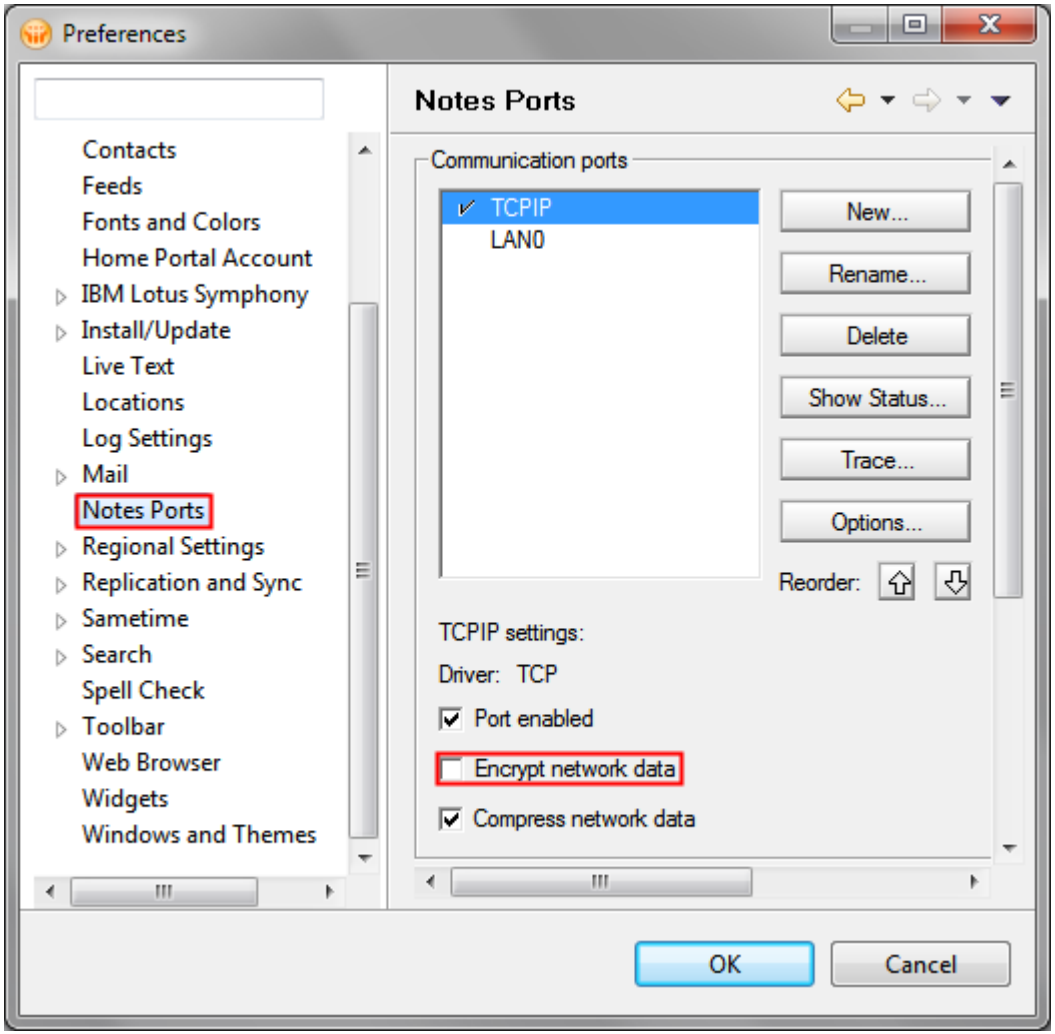
- In Notes and Domino you can encrypt
  - network traffic,
  - databases and
  - fields.
- In- and outgoing emails are encrypted using the field encryption.

Network encryption

- Network traffic can be encrypted.
- If **at least one side** – the client or the server – wants to encrypt, the connection will be encrypted.
- Network encryption is **configured per port**.
- RC4 is used for network encryption.

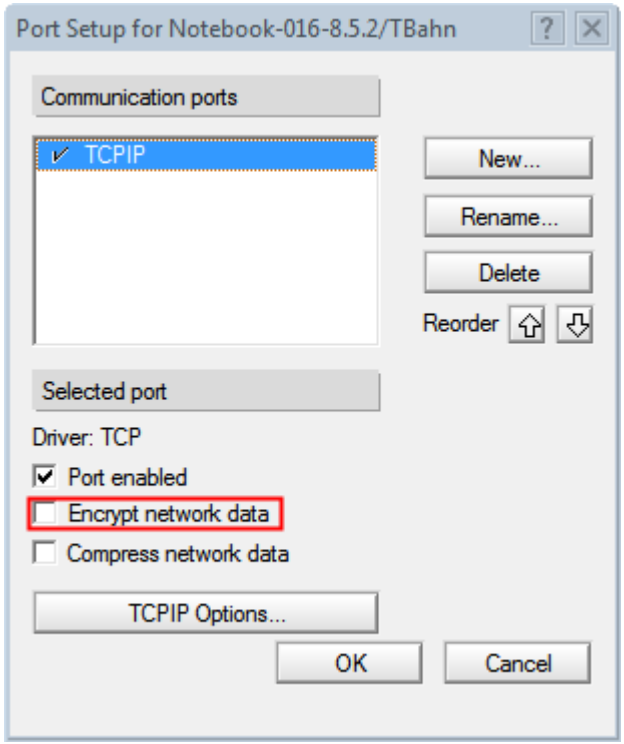
Enabling Network Encryption for Notes Client

- File – Preferences...



Enabling Network Encryption for Domino Server

- in the Domino Administrator
  Server... – Tools – Ports – Setup...

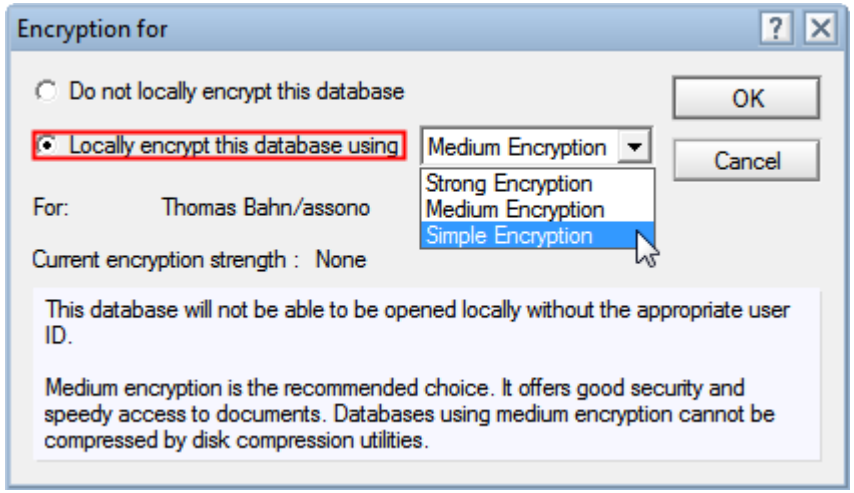


Encryption of Databases

- Notes databases, i.e. nsf and ntf files, can be stored encrypted in the file system.
- On the server, its public key is used to encrypt the database, on the client the selected user's public key.
- This way, only the owner of the private key can decrypt and use the information in the database.

Encryption of Databases

- There are 3 levels:
  - Strong Encryption
  - Medium Encryption
  - Simple Encryption

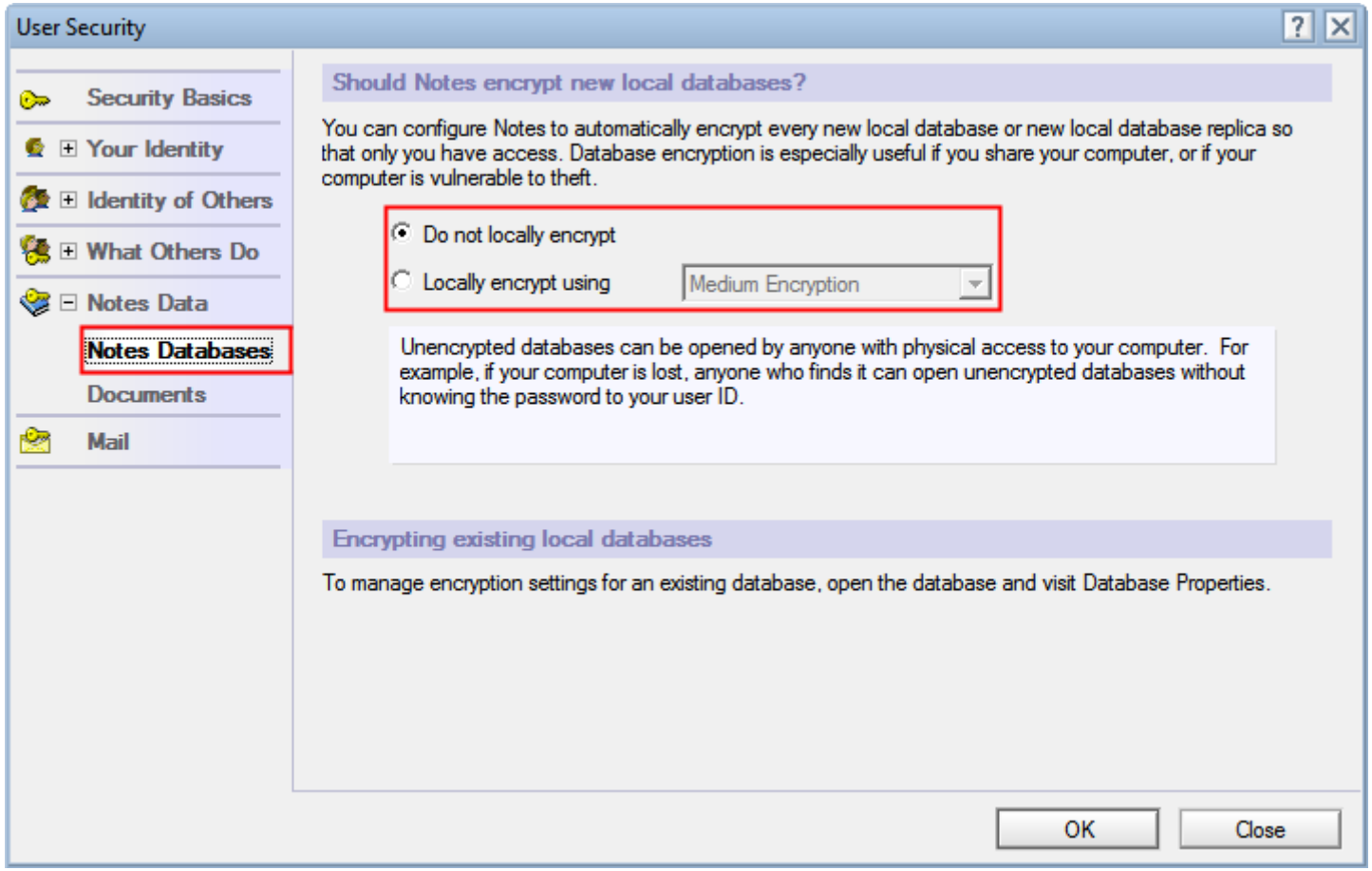


- Higher levels are more secure, but cost more CPU time and are slower.

- You can choose to encrypt a database, when you create a new one, when you create a new replica or later. In this case you must compact the database to enable the encryption.

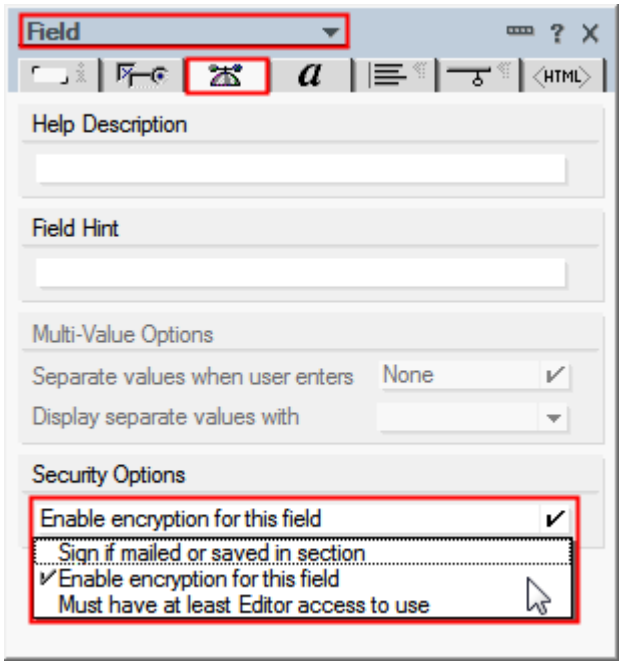
Control the Default for New Databases

- Under Security – User Security... you can control the default for new local databases and replicas.



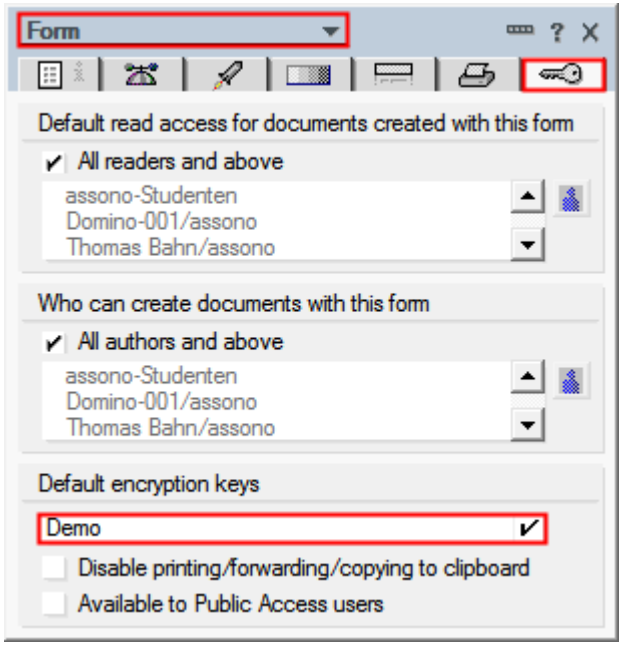
Encryption of Fields in Documents

- Notes developers can **set for each field** in a form that the corresponding item should be stored encrypted.
- For RichText items, the attached files are stored encrypted, too.
- For password fields, this property is set automatically.



Encryption of Fields in Documents (cont.)

- Additionally one or more keys have to be set in the form's properties.
- You can choose to use:
  - public keys from users or
  - secret keys



Public Keys From Users

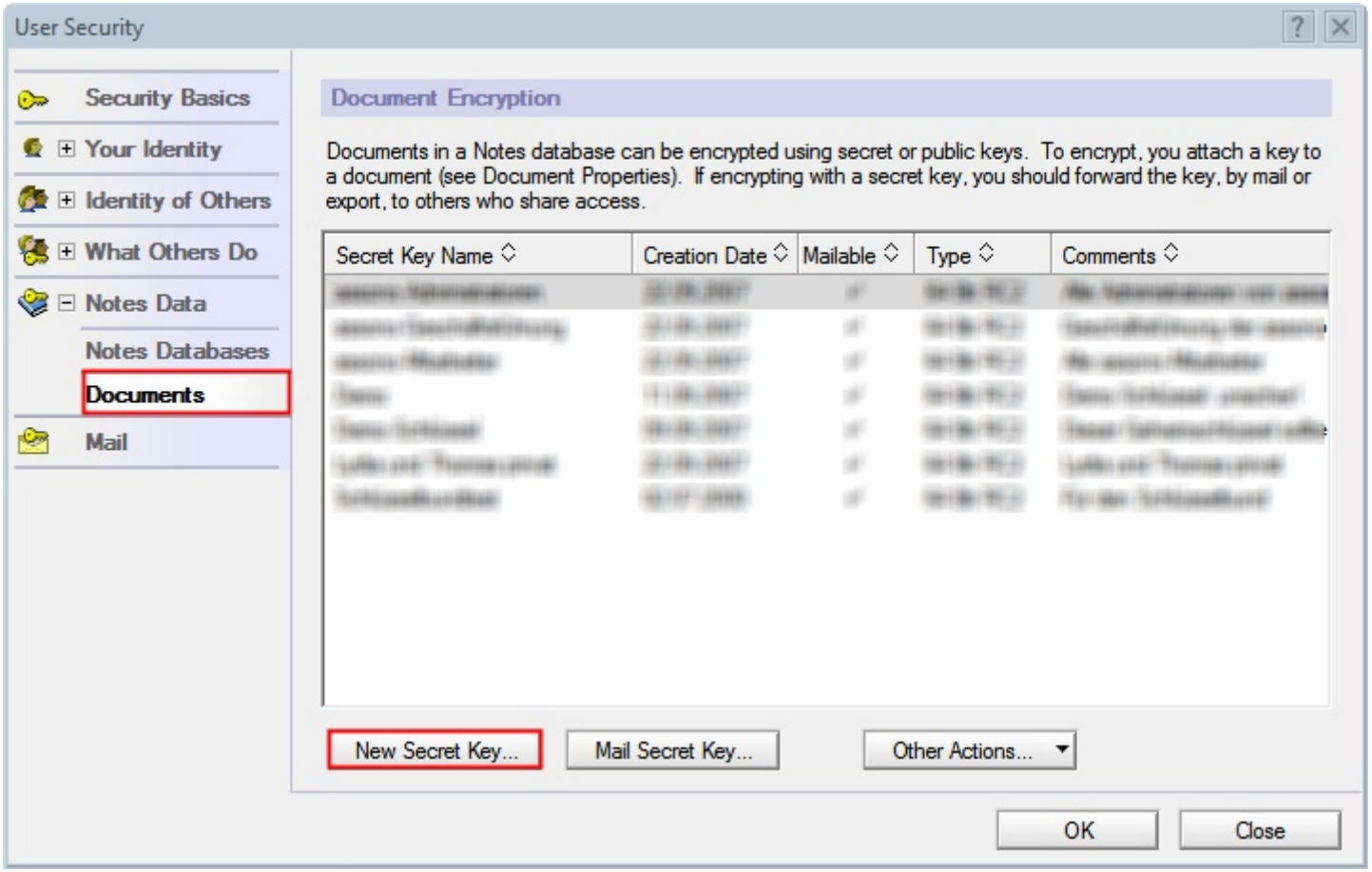
- The developer must create a Names field with the name “PublicEncryptionKeys”.
- The item should contain the NotesNames of all persons, the document should be usable for.
- When a document is saved or send, the public keys of the named users are looked up in the Domino Directory.
- Then all marked items are encrypted using a random key and the RC2 algorithm. The random key is stored encrypted once for each person.

Secret Keys

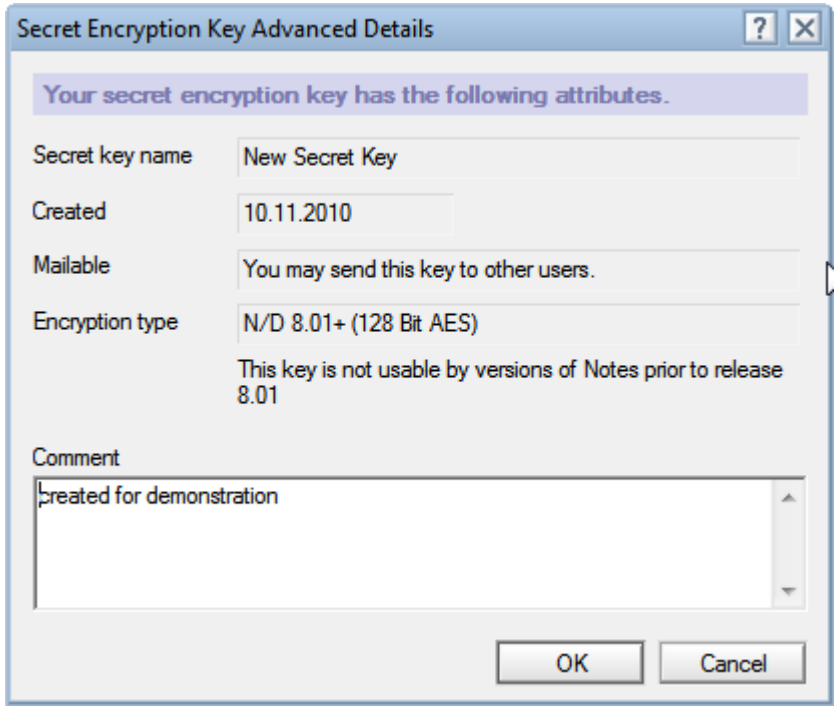
- So called secret keys can be stored in ID files.
- In the form's properties, the developer can choose a default secret key from those stored in his ID file.
- If a form contains an item SecretEncryptionKeys, it should contain the name of a secret key.
- Else this item is created from the form's default.
- Secret keys can be
  - created in an ID file,
  - exported from it,
  - send by email and
  - imported into other ID files.

Check All Your Secret Keys

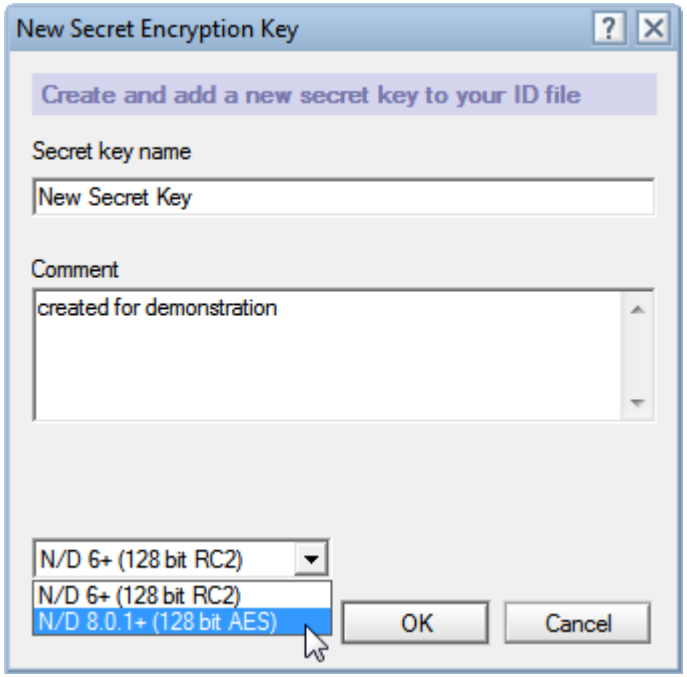
- You can see all your secret keys in Security – User Security...



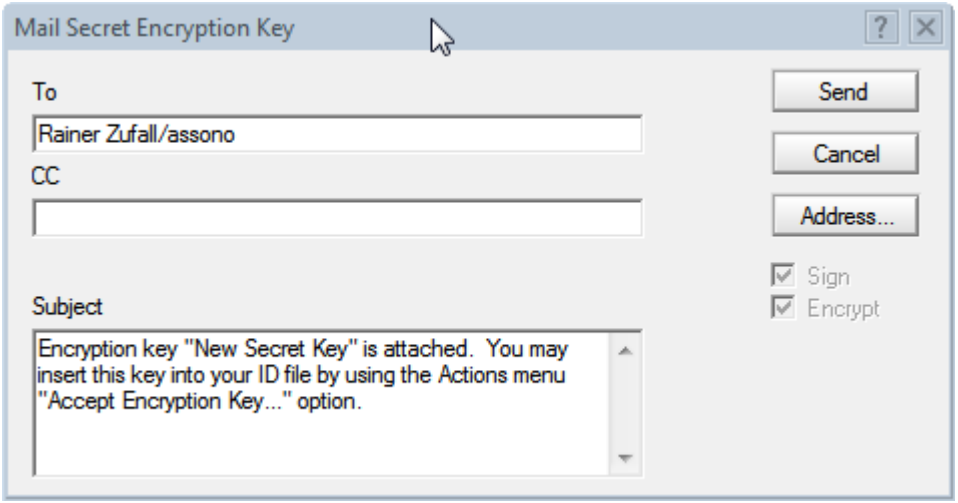
Examine a Secret Key's Details



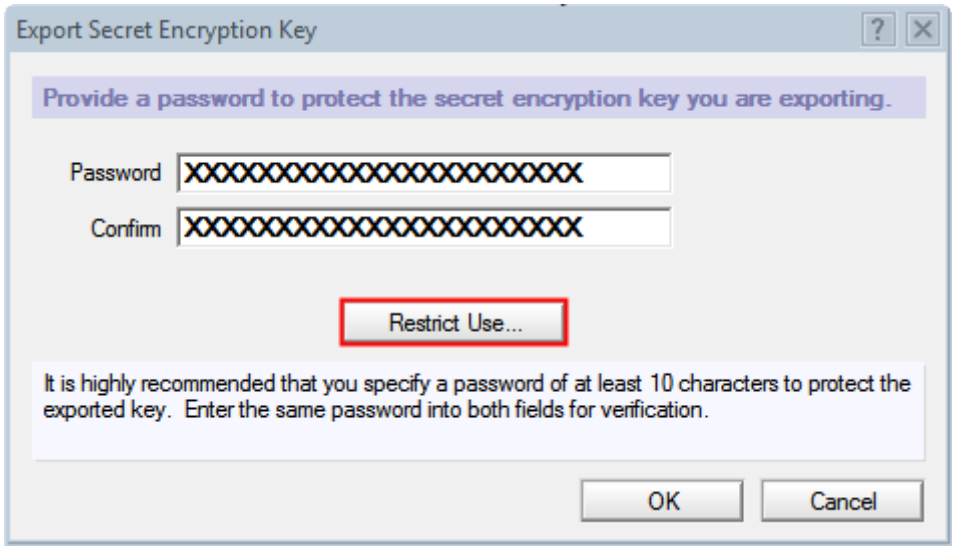
Create a New Secret Key

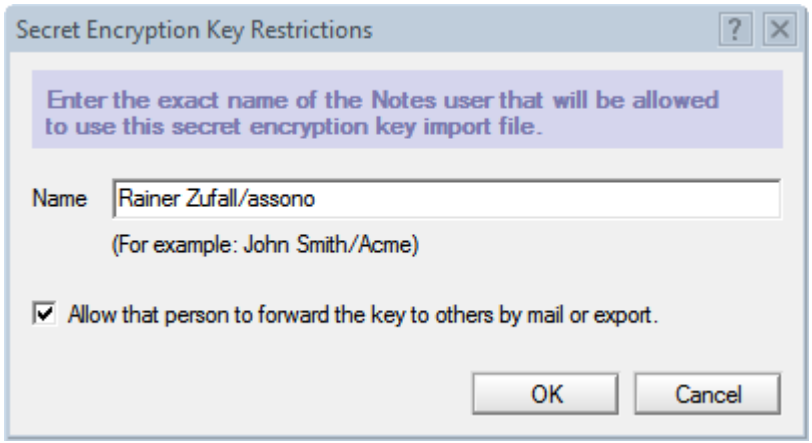


Then Send It per Email



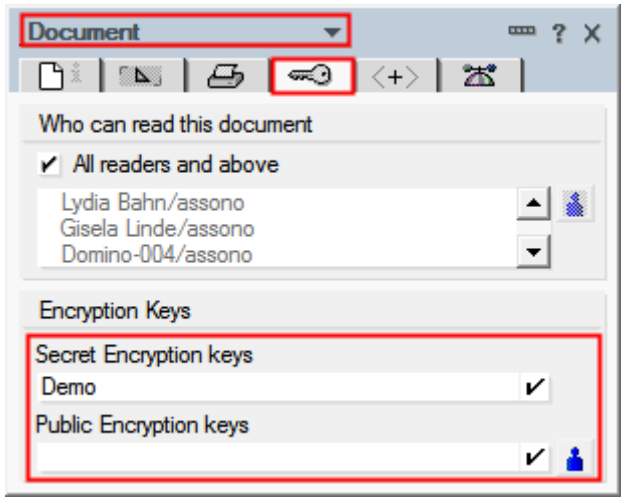
Or Export It to a File





There is Something More to Mention...

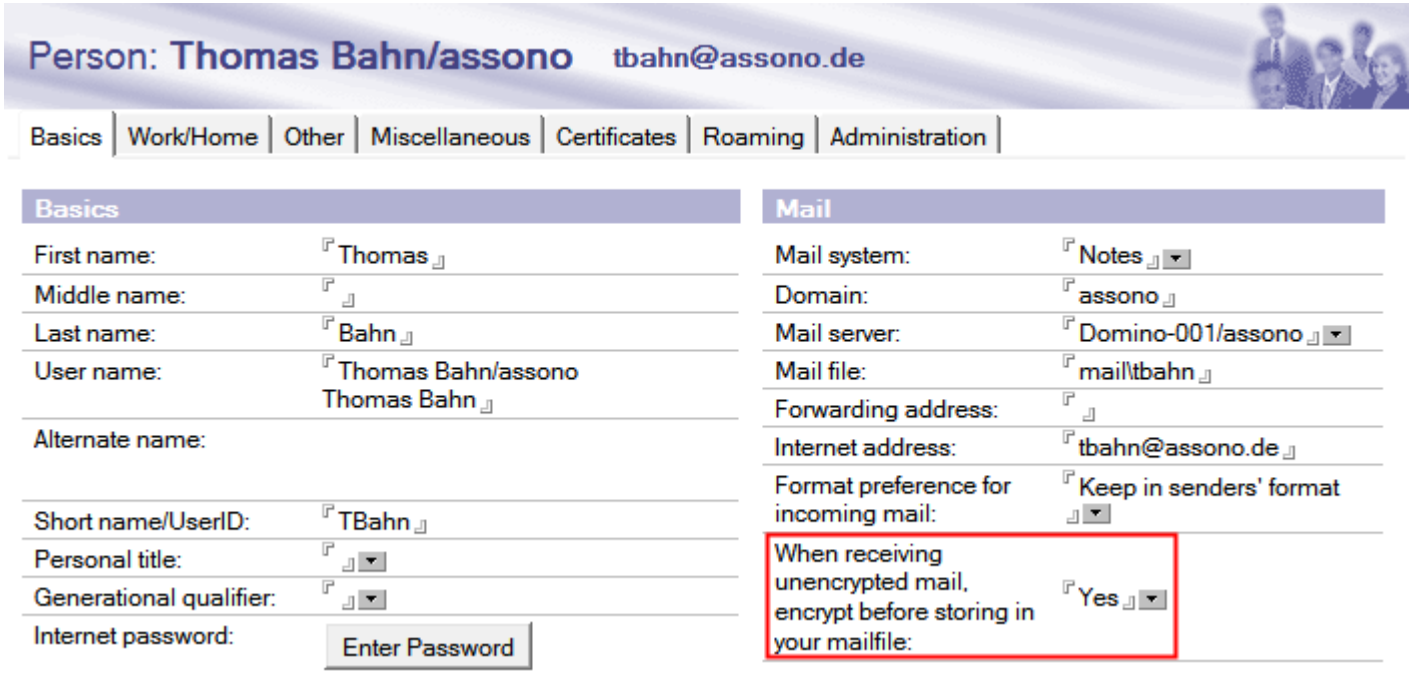
- If a form has some fields flagged to be stored encrypted, the user can choose encryption keys in any document using this form in the document's properties.



- Combine this with each user's capability to create new secret keys, stored only in their own ID file, protected by their password...
- If not already in place, really think about setting up ID recovery or ID vault.

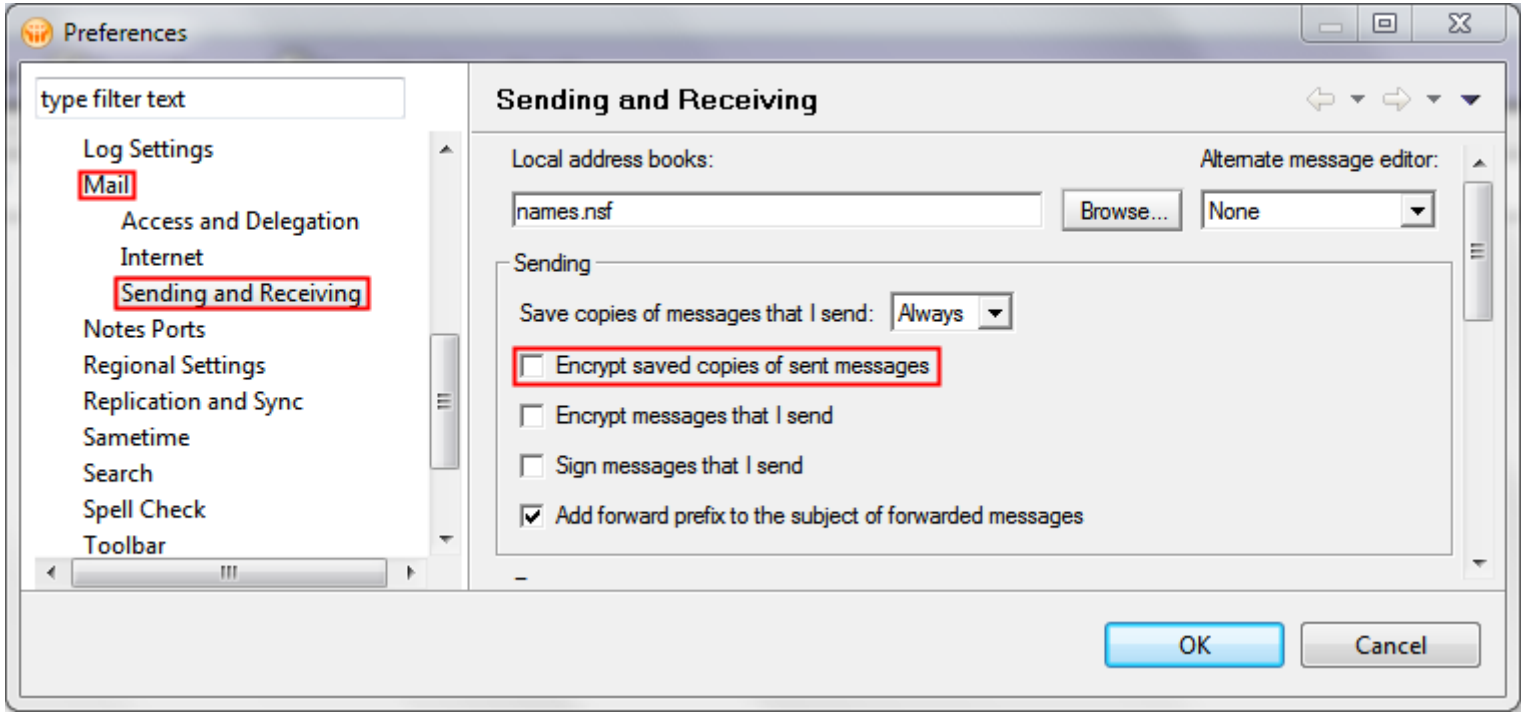
Email Encryption (Notes-internal)

- In- and outgoing emails can be encrypted.
- For incoming emails, this can be controlled the person's document in the Domino Directory:



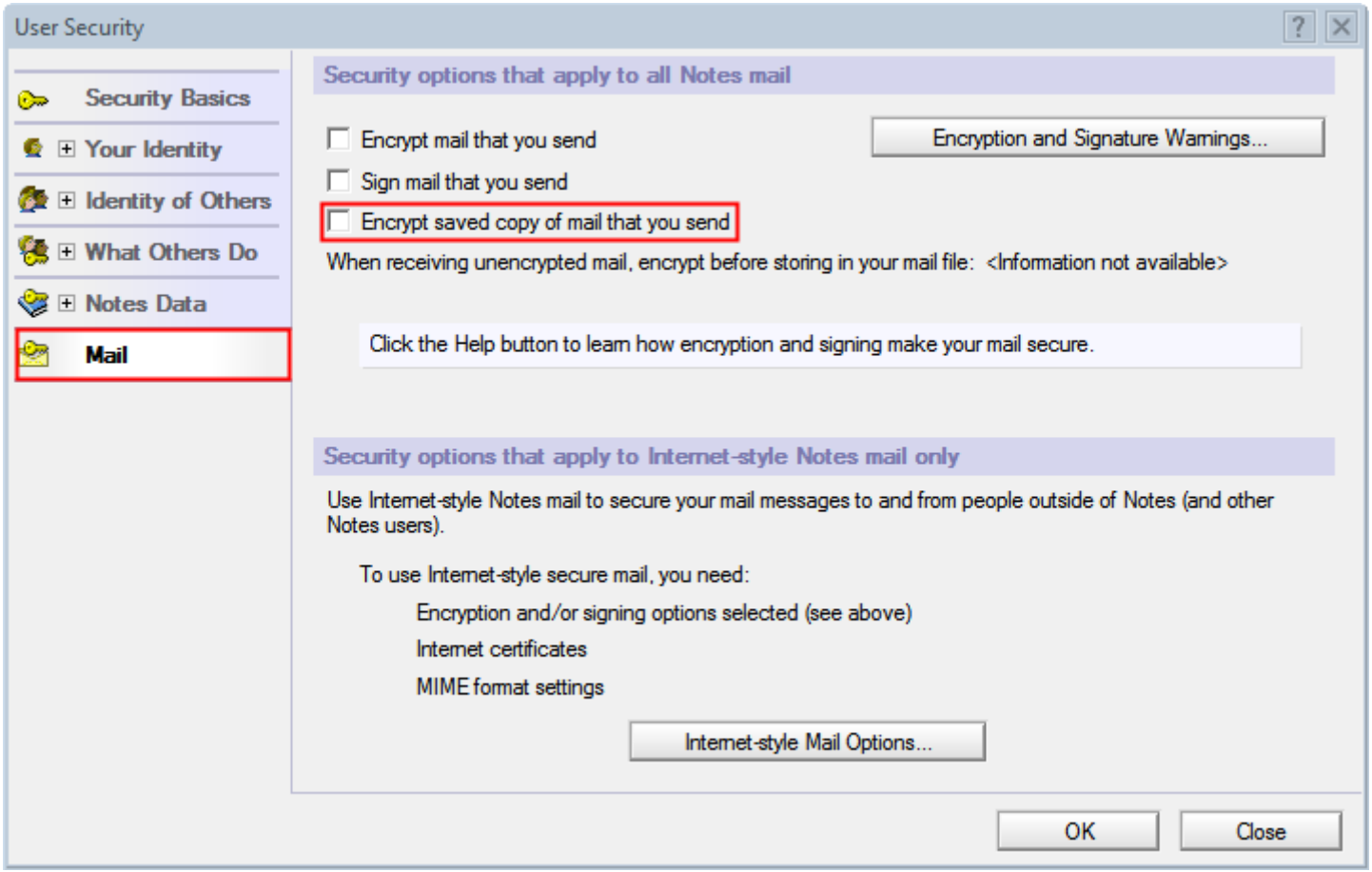
Encryption of Stored Outgoing Emails

- When sending emails, a copy of the email can be stored encrypted.
- This can be set up in the client's mail preferences:



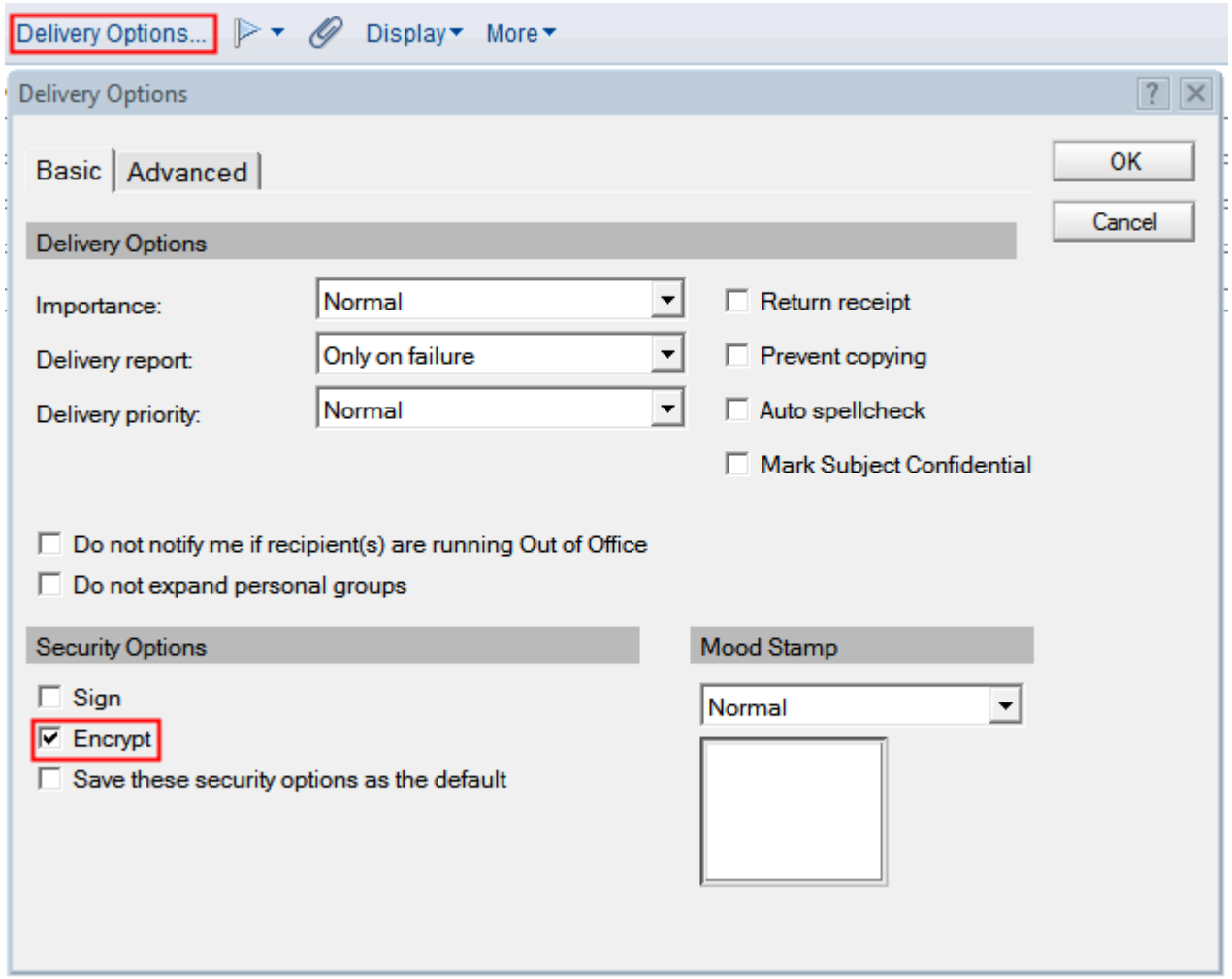
Encryption of Stored Outgoing Emails (cont.)

- Or you can use Security – User Security...:



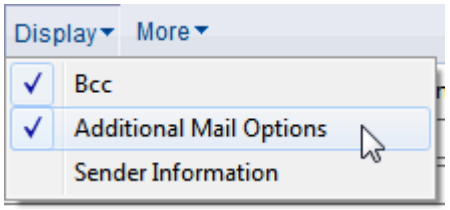
Encryption of Outgoing Emails

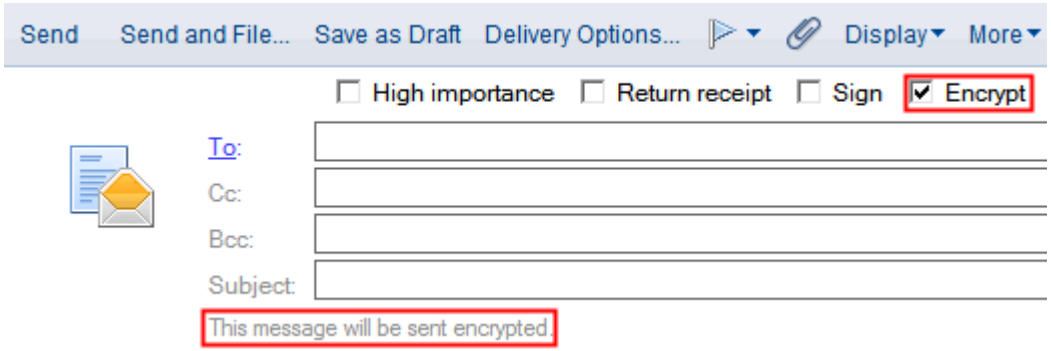
- For each outgoing email, the user can switch on its encryption in the Delivery Options...:



Encryption of Outgoing Emails (cont.)

- Or he can switch on the Advanced Mail Options and select Encrypt at the top:





Agenda

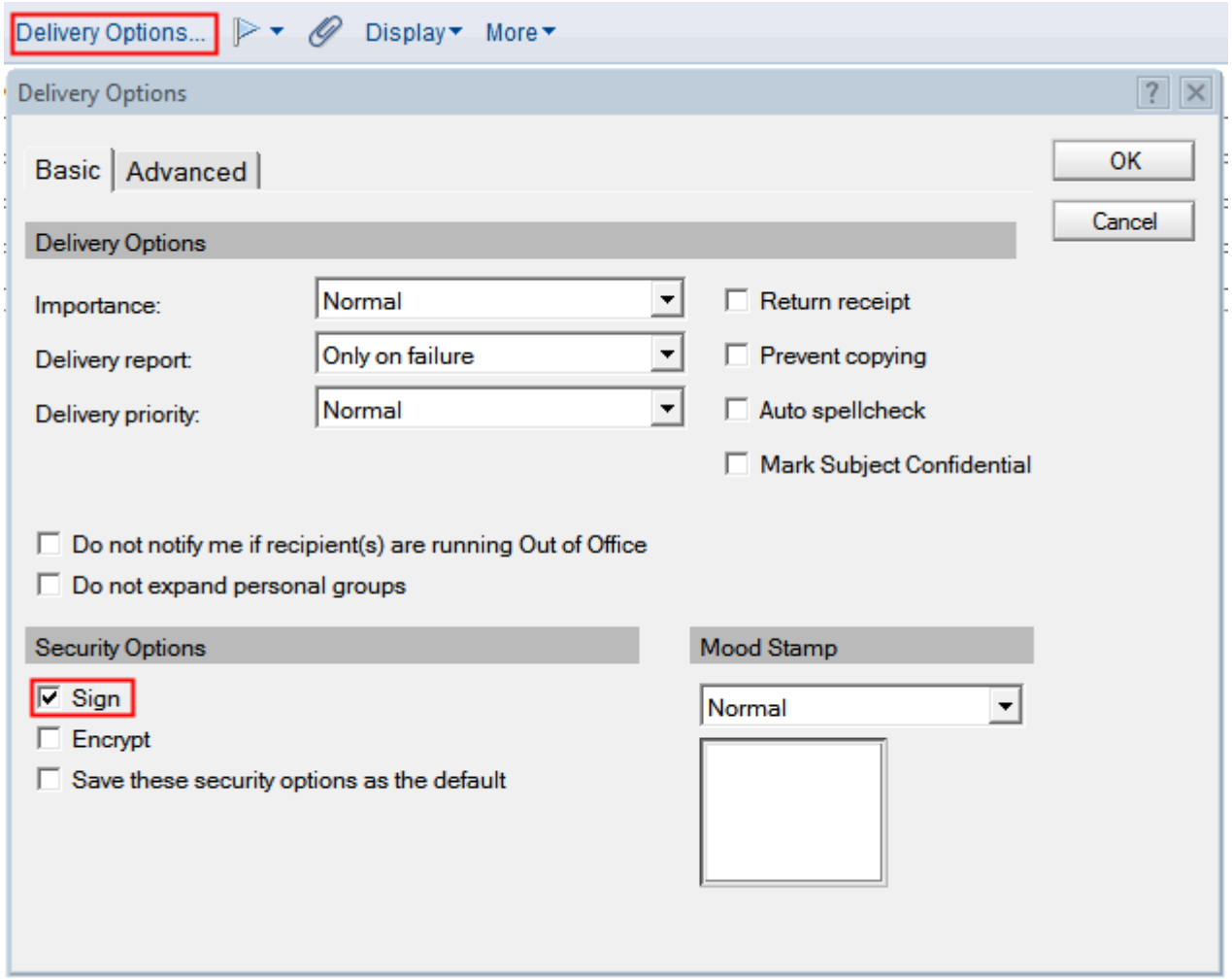
- Modern Cryptography – The Basics
  - Encryption & Decryption
  - Hash Functions and Electronic Signatures
- Notes and Domino
  - Certificates and ID files
  - Encryption & Decryption
  - **Electronic Signatures**
  - Internet

Signatures in Notes and Domino

- Signatures in Notes and Domino can be applied to:
  - outgoing emails
  - documents
  - controlled-access sections

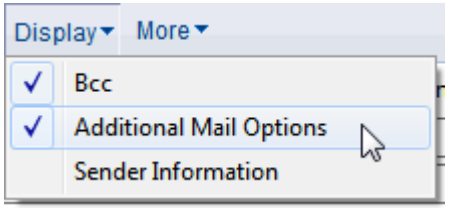
Singing Outgoing Emails

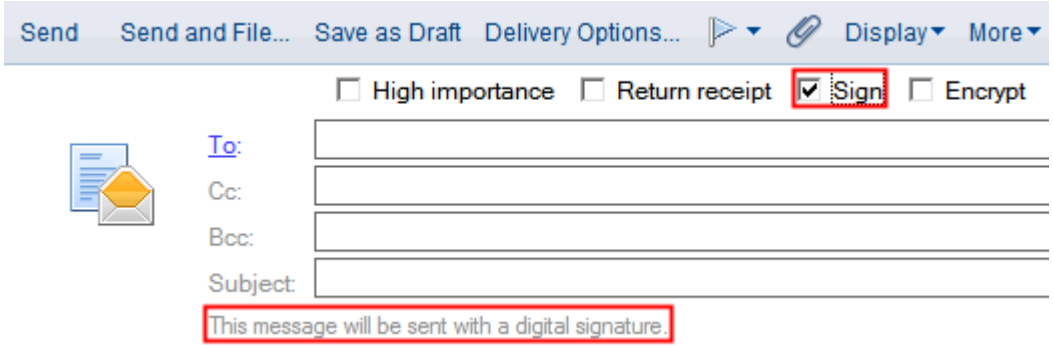
- For each outgoing email, the user can set it to be signed in the Delivery Options...:



Singing Outgoing Emails (cont.)

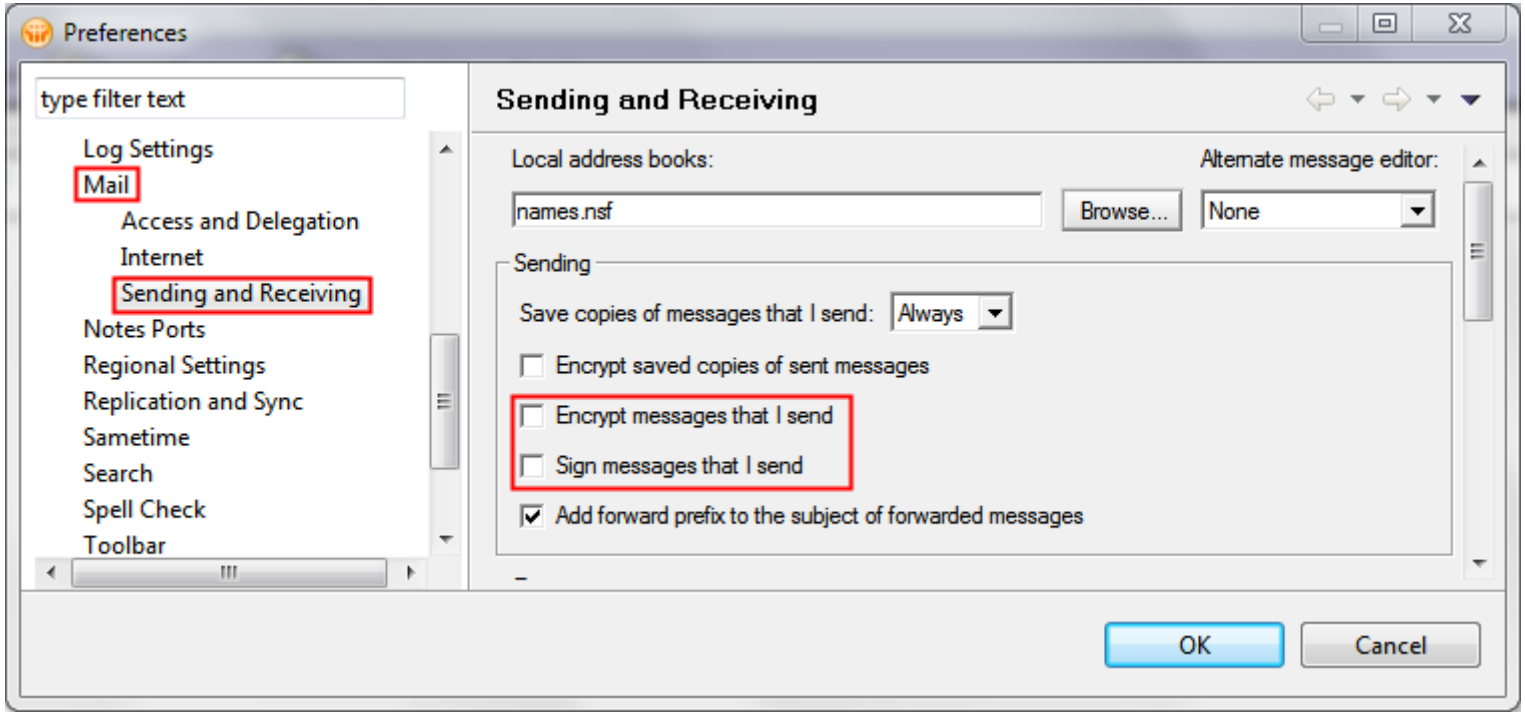
- Or he can switch on the Advanced Mail Options and select Sign at the top:





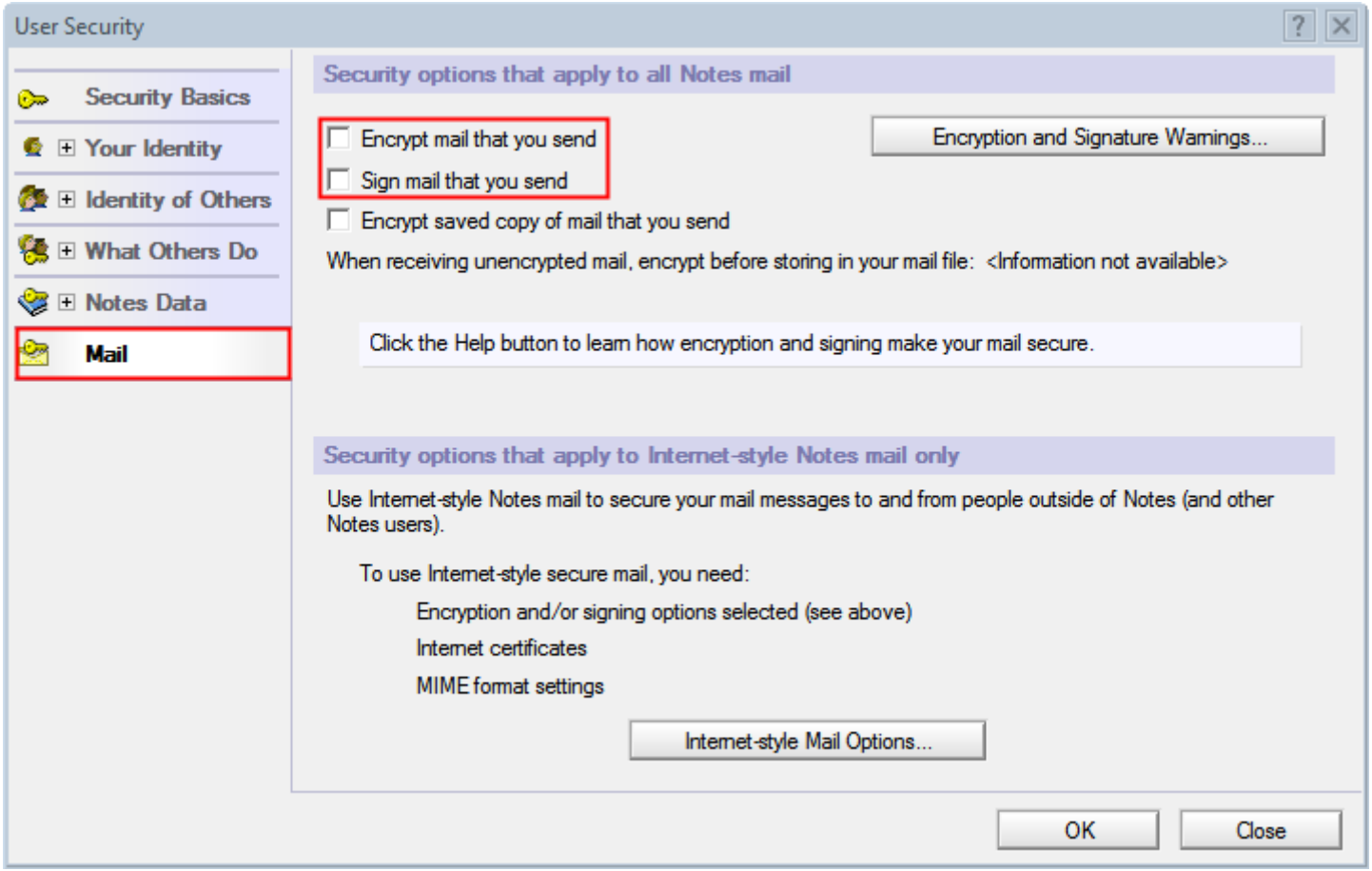
Setting Defaults For Encryption and Signing

- In the client's mail preferences, the user can set the defaults for encryption and signing:



Setting Defaults For Encryption and Signing (cont.)

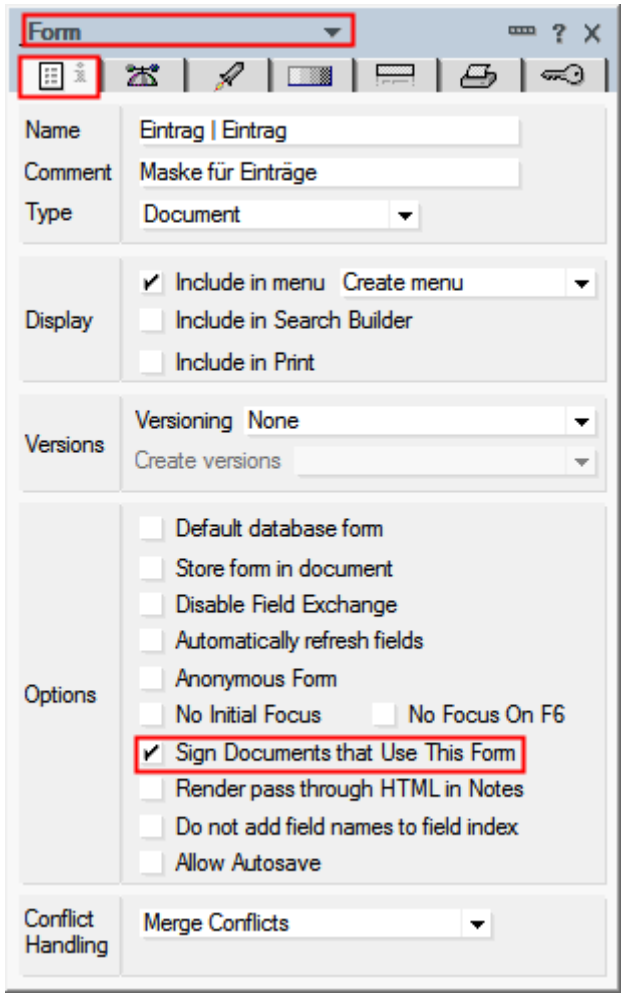
- Or he can use Security – User Security...:



Signing Documents

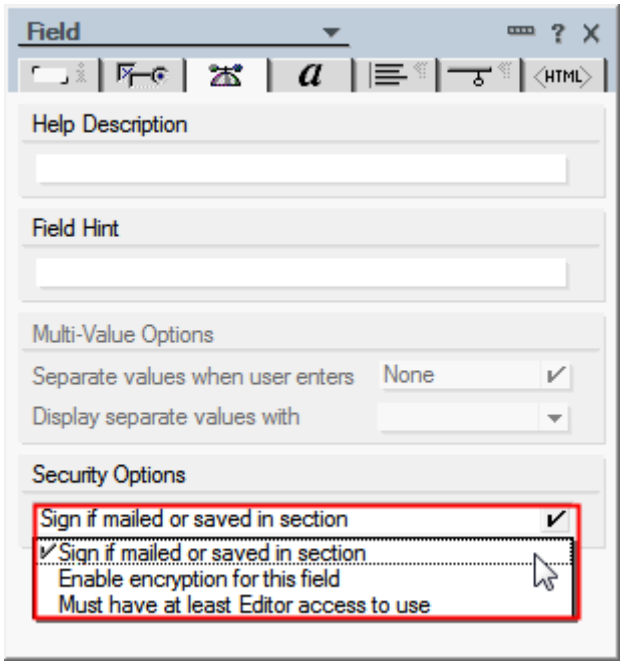
- Notes developers can set a form's property to sign all documents saved or send using this form.
- When a signed document is opened, the details are shown in the status bar:

Signed by Thomas Bahn/assono on 19.04.2010 10:46:56, according to /assono



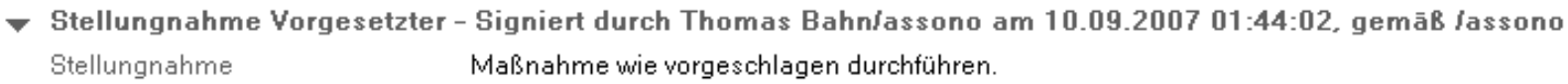
Signing Controlled-Access Sections

- If a controlled-access section contains at least one field with its Security Options containing "Sign if mailed or saved in section", this section will be signed, when the document is saved or send.



- The signatures of all sections are updated later, if the current user has write access to their contents.

- When a document with a signed section is opened, the details are shown beside the section's title:



Agenda

- Modern Cryptography – The Basics
  - Encryption & Decryption
  - Hash Functions and Electronic Signatures
- Notes and Domino
  - Certificates and ID files
  - Encryption & Decryption
  - Electronic Signatures
  - **Internet**

HTTP + SSL = HTTPS

- **Secure Sockets Layer** (**SSL**) and **Transport Layer Security** (**TLS**) are networking protocols for the secure transport of data over the insecure internet.
- HTTP and SSL together are called HTTPS and used to provide security for Web applications through encryption.

HTTP + SSL = HTTPS (cont.)

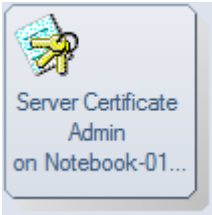
- HTTPS authentication works like authentication in Notes and Domino:
  - The Server has a certificate signed by a CA.
  - The CA is a trusted third party (and you have to pay for the certification) or you create it yourself.
  - Browsers know some important trust centers, i.e. CAs, which certificates are built-in the browser.
  - You can import other CAs into the browser, including the ones, you created yourself.

HTTP + SSL = HTTPS (cont.)

- When a HTTPS protected site is opened, the browser checks its certificate.
- If it cannot validate the server's certificate, it asked the user, who can stop, trust it once or for ever.
- By importing your self-signed certificates before, you can prevent this confusing choice for your users.
- This is only the first half the the Notes/Domino authentication process, after which the client knows and trusts the server.
- The network traffic is encrypted using a random key and a symmetric algorithm.

Creating Server SSL Certificates

- You can create SSL certificates using the Server Certificate Admin database (see Admin help):



SSL-related Settings in the Server Document

Server: **Notebook-016-8.5.2/TBahn** notebook-016-86

Basics | Security | Ports... | Server Tasks... | Internet Protocols... | MTAs... | Miscellaneous

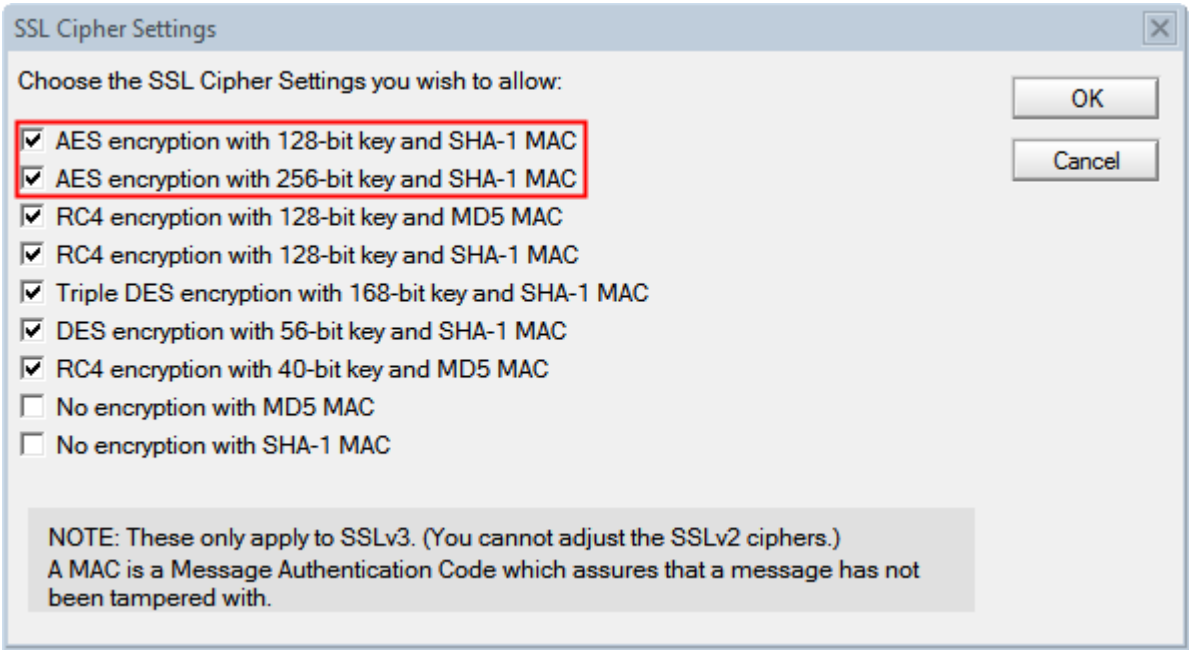
Notes Network Ports | Internet Ports... | Proxies

SSL settings

SSL key file name:	keyfile.kyr
SSL protocol version (for use with all protocols except HTTP):	Negotiated
Accept SSL site certificates:	○ Yes ◉ No
Accept expired SSL certificates:	◉ Yes ○ No
SSL ciphers: Modify	RC4 encryption with 128-bit key and MD5 MAC RC4 encryption with 128-bit key and SHA-1 MAC Triple DES encryption with 168-bit key and SHA-1 MAC DES encryption with 56-bit key and SHA-1 MAC RC4 encryption with 40-bit key and MD5 MAC AES encryption with 128-bit key and SHA-1MAC AES encryption with 256-bit key and SHA-1 MAC
Enable SSL V2: (SSL V3 is always enabled)	☐ Yes

Changing SSL Cipher Settings

- By default, AES encryption is disabled.



SSL Client Certificates

- You can also create a SSL certificate for a user.
- This is like the certificates stored in the Notes ID files.
- Normally these client certificates are protected by a password, too.
- A (Domino) Web server can be configured to accept browsers showing a SSL client certificate – in addition to user/password or exclusively.
- Two-factor authentication (file and password) is much more secure, but you have to maintain the client certificates.

Secure Internet Emails Using S/MIME

- MIME (Multipurpose Internet Mail Extensions): Standard for structure and sending of “rich” emails in the Internet.
- MIME-encoded emails can contain multiple section with e.g. a plain text and a HTML version of the mail body and additional ones for each attachment and embedded image.
- Domino can send MIME-encoded emails.

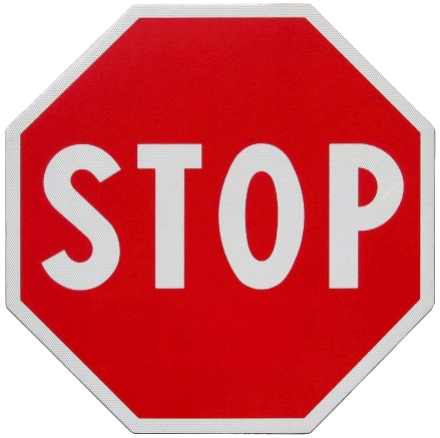
Secure Internet Emails Using S/MIME (cont.)

- S/MIME (Secure/Multipurpose Internet Mail Extensions): Standard for encrypting and signing emails in the Internet.
- Works like the Notes-internal encryption. 
- MIME sections are encrypted with the recipients public key and signed the the senders private key.
- The certificate of the sender is attached to all signed emails.

Resources

- Administrator Help: contains a lot of step-by-step instructions and explanations
- IBM Redbooks und Redpapers (oldies, but goldies) 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)

Questions?



Ask questions now — or later:

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

assono
IT-Consulting & Solutions

Presentation will be posted at:
www.assono.de/blog/d6plinks/ILUG-2010-Cryptography