System z Expo

October 13 – 17, 2008 – Las Vegas, Nevada

Session Title: z/VSE Security Exploitation with Crypto Hardware

Session ID: zES02

Speaker Name: Ingo Franzki

Authorized IBM Training

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

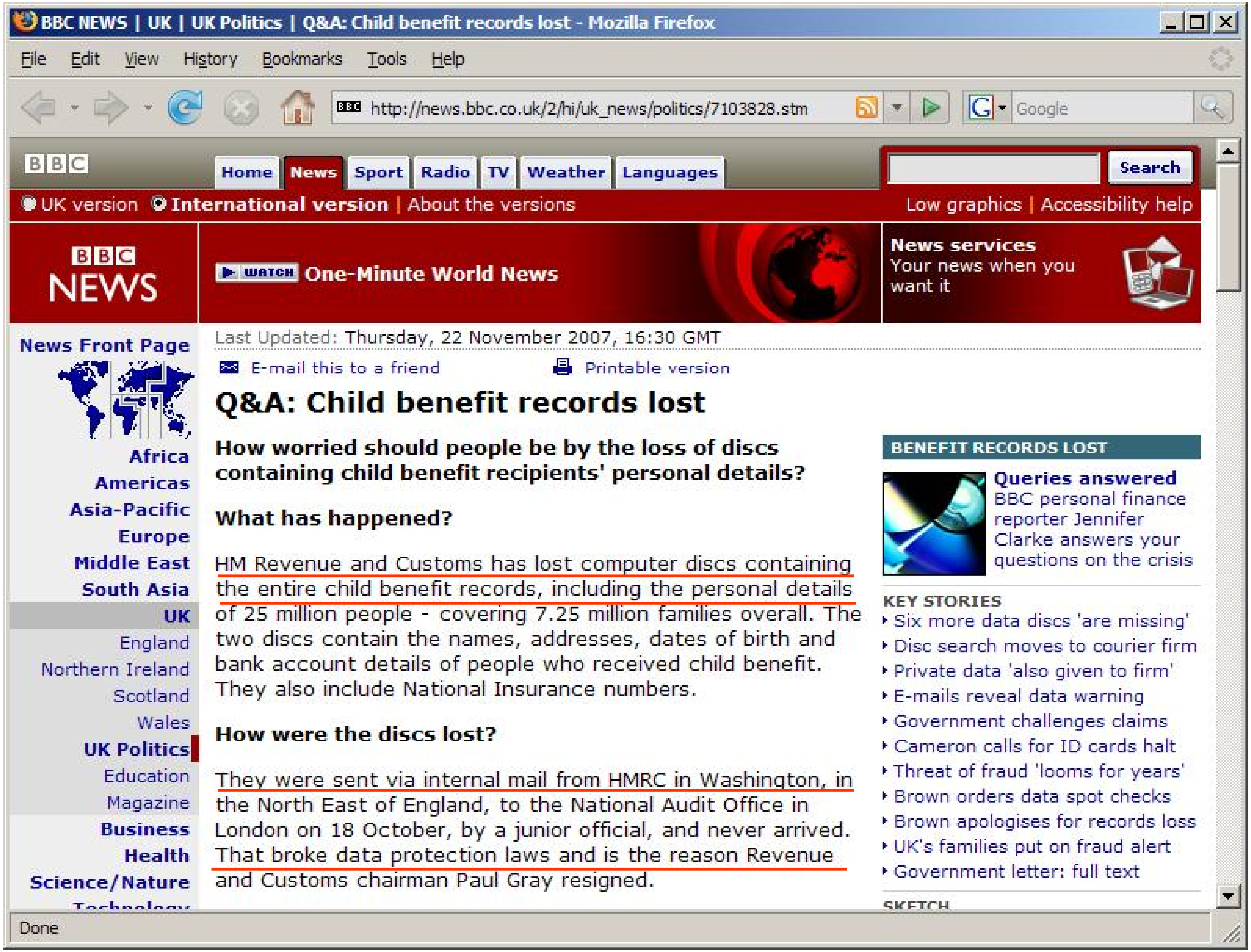
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Security requirements

- Security requirements are increasing in today's world
  - Data security
  - Data integrity
  - Keep long-term data audit-save
- The number of attacks increase daily
  - Industrial spying
  - Security exploits, Denial-of-Service attacks
  - Spam, Phishing, …
- Not paying attention to security requirements can be very expensive
  - Your data is the heart of your company
  - Loosing your customer data is a disaster
  - You can loose customers
- IT Security gets more and more important
  - You need to consider the whole IT Environment not only single systems


BBC NEWS | UK | UK Politics | Q&A: Child benefit records lost - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://news.bbc.co.uk/2/hi/uk_news/politics/7103828.stm
Google
BBC
Home News Sport Radio TV Weather Languages
Search
UK version International version | About the versions
Low graphics | Accessibility help
BBC NEWS
WATCH One-Minute World News
News services
Your news when you want it
News Front Page
Africa
Americas
Asia-Pacific
Europe
Middle East
South Asia
UK
England
Northern Ireland
Scotland
Wales
UK Politics
Education
Magazine
Business
Health
Science/Nature
Technology
Done

Last Updated: Thursday, 22 November 2007, 16:30 GMT

E-mail this to a friend   Printable version

Q&A: Child benefit records lost

How worried should people be by the loss of discs containing child benefit recipients' personal details?

What has happened?

HM Revenue and Customs has lost computer discs containing the entire child benefit records, including the personal details of 25 million people - covering 7.25 million families overall. The two discs contain the names, addresses, dates of birth and bank account details of people who received child benefit. They also include National Insurance numbers.

How were the discs lost?

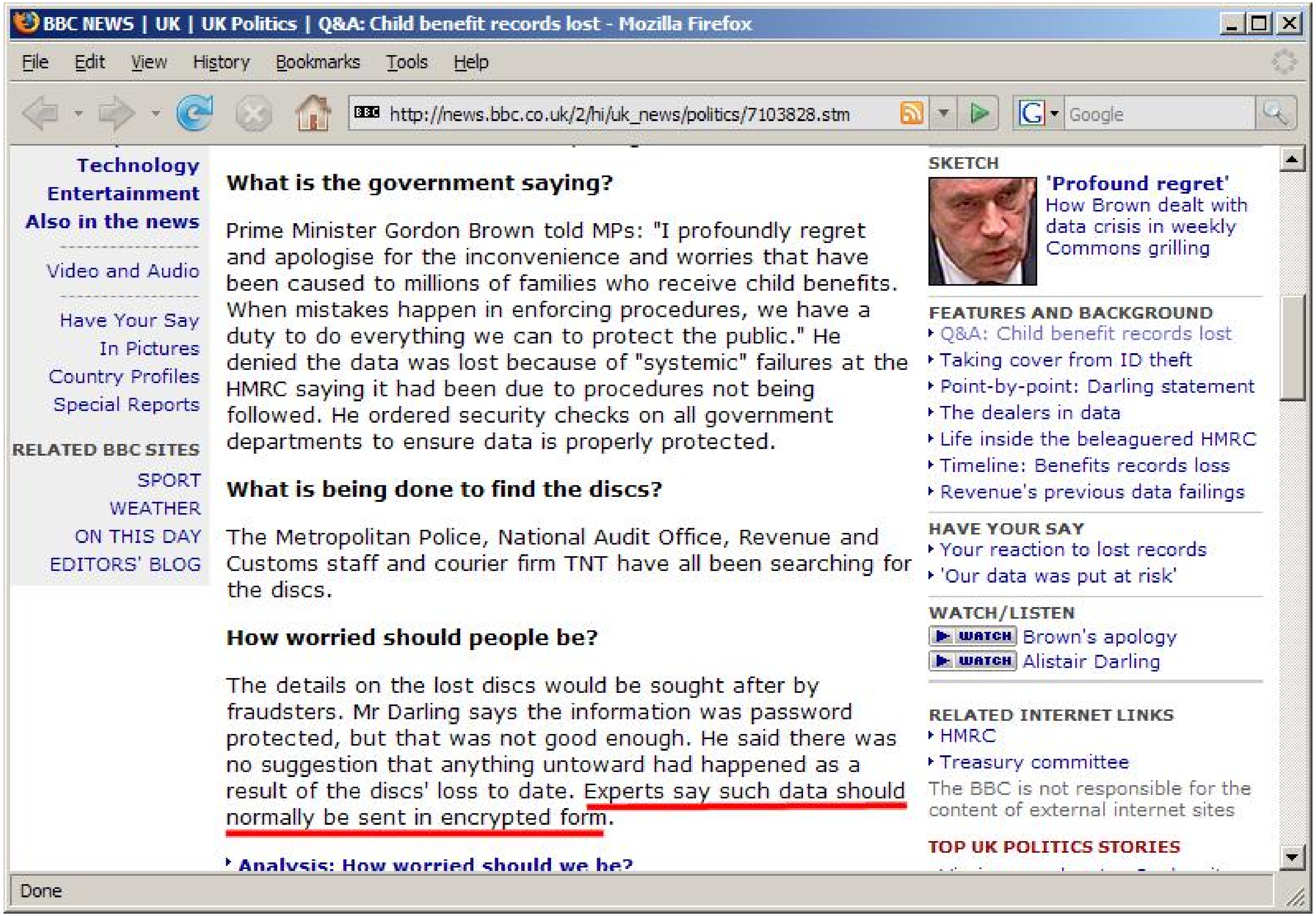
They were sent via internal mail from HMRC in Washington, in the North East of England, to the National Audit Office in London on 18 October, by a junior official, and never arrived. That broke data protection laws and is the reason Revenue and Customs chairman Paul Gray resigned.

BENEFIT RECORDS LOST

Queries answered
BBC personal finance reporter Jennifer Clarke answers your questions on the crisis

KEY STORIES
- Six more data discs 'are missing'
- Disc search moves to courier firm
- Private data 'also given to firm'
- E-mails reveal data warning
- Government challenges claims
- Cameron calls for ID cards halt
- Threat of fraud 'looms for years'
- Brown orders data spot checks
- Brown apologises for records loss
- UK's families put on fraud alert
- Government letter: full text

SKETCH



What is the government saying?

Prime Minister Gordon Brown told MPs: "I profoundly regret and apologise for the inconvenience and worries that have been caused to millions of families who receive child benefits. When mistakes happen in enforcing procedures, we have a duty to do everything we can to protect the public." He denied the data was lost because of "systemic" failures at the HMRC saying it had been due to procedures not being followed. He ordered security checks on all government departments to ensure data is properly protected.

What is being done to find the discs?

The Metropolitan Police, National Audit Office, Revenue and Customs staff and courier firm TNT have all been searching for the discs.

How worried should people be?

The details on the lost discs would be sought after by fraudsters. Mr Darling says the information was password protected, but that was not good enough. He said there was no suggestion that anything untoward had happened as a result of the discs' loss to date. Experts say such data should normally be sent in encrypted form.

- Analysis: How worried should we be?

SKETCH

'Profound regret'
How Brown dealt with data crisis in weekly Commons grilling

FEATURES AND BACKGROUND

- Q&A: Child benefit records lost
- Taking cover from ID theft
- Point-by-point: Darling statement
- The dealers in data
- Life inside the beleaguered HMRC
- Timeline: Benefits records loss
- Revenue's previous data failings

HAVE YOUR SAY

- Your reaction to lost records
- 'Our data was put at risk'

WATCH/LISTEN

- WATCH Brown's apology
- WATCH Alistair Darling

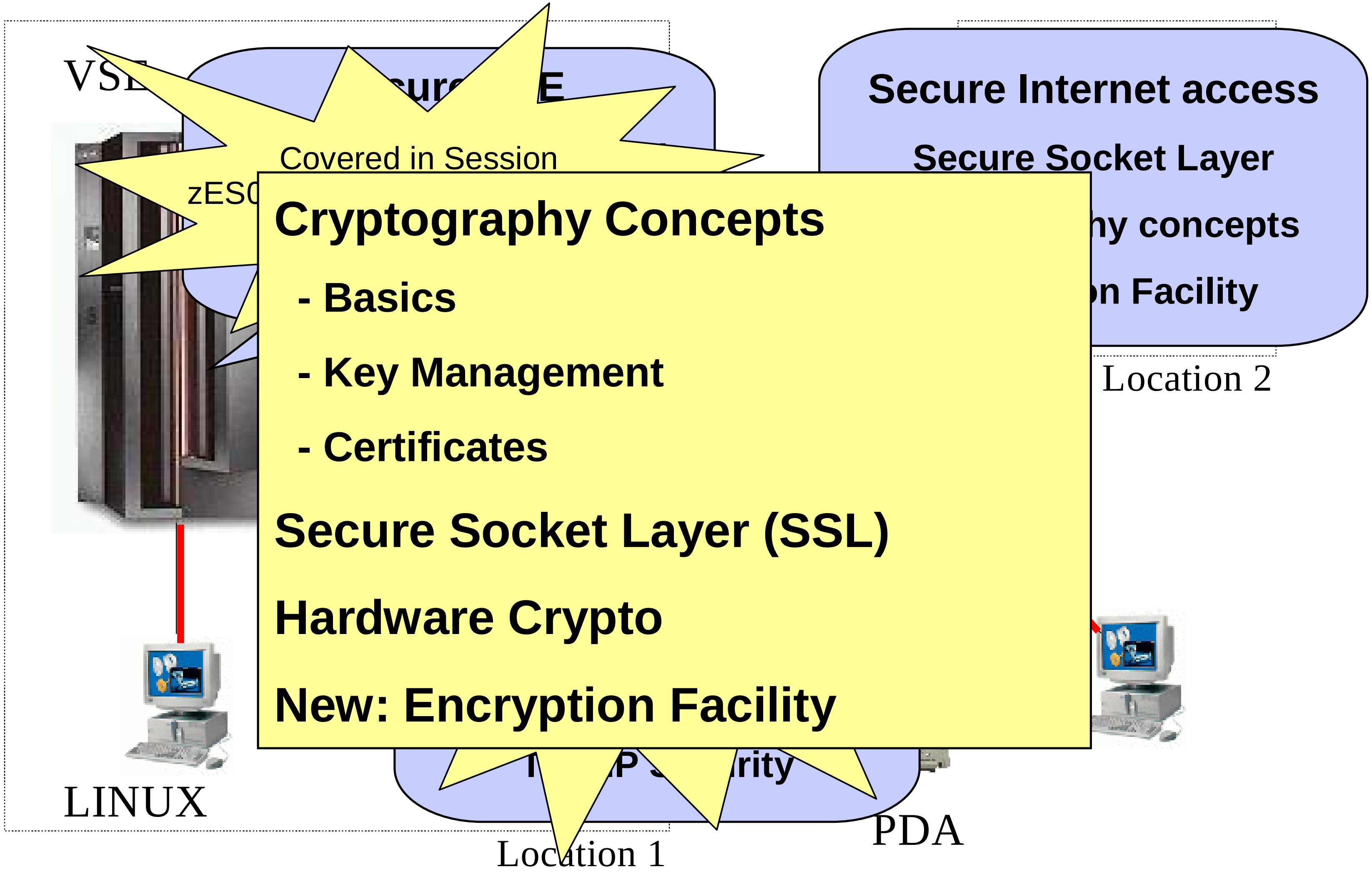
RELATED INTERNET LINKS

- HMRC
- Treasury committee

The BBC is not responsible for the content of external internet sites

TOP UK POLITICS STORIES

Security in a heterogeneous environment



Covered in Session

Cryptography Concepts

- **Basics**
- **Key Management**
- **Certificates**

Secure Socket Layer (SSL)

Hardware Crypto

New: Encryption Facility

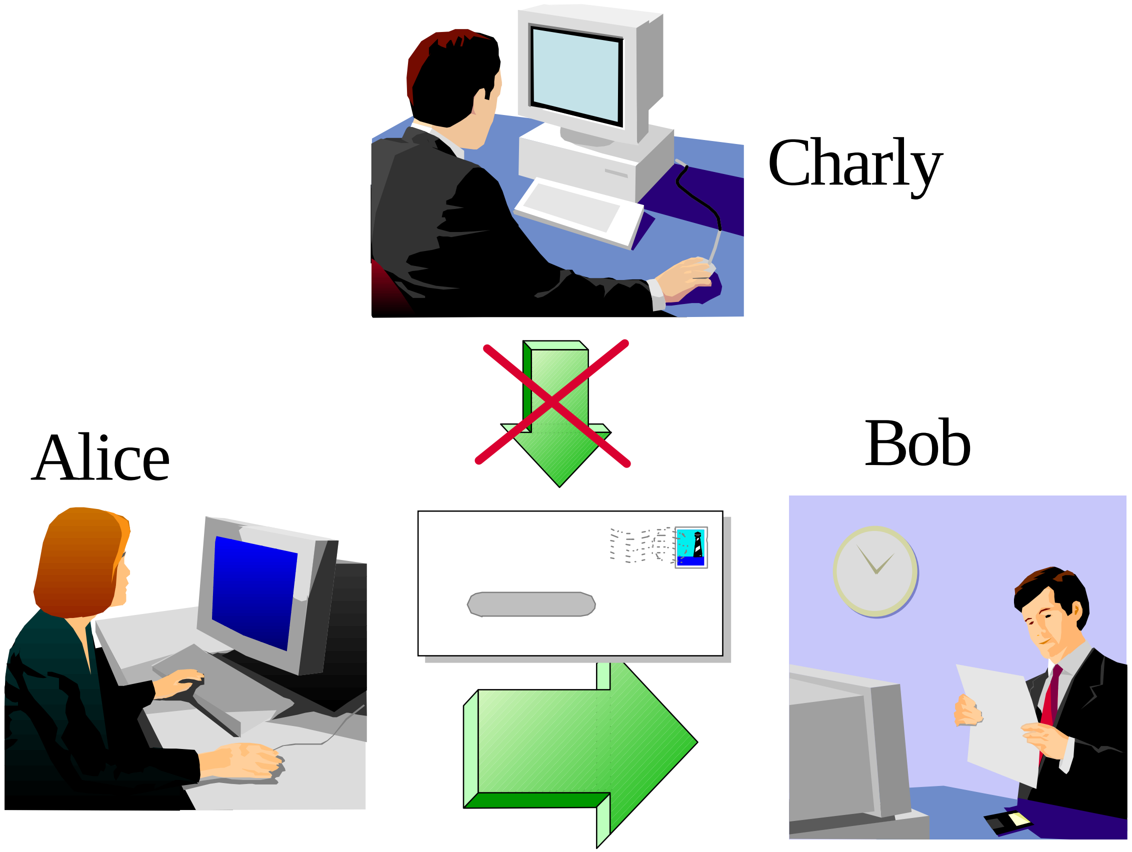
What can cryptography do for you?

- 2 main areas
  - Encryption of data transmitted over TCP/IP connections
    - SSL, HTTPS
    - SecureFTP
  - Encryption of data stored on disk or tape
    - Encryption of backups or archives
    - Exchange of encrypted and/or signed data with customers or business partners
    - TS1120 Encrypting Tape Drive
    - Encryption Facility for z/VSE

Why Cryptography ?

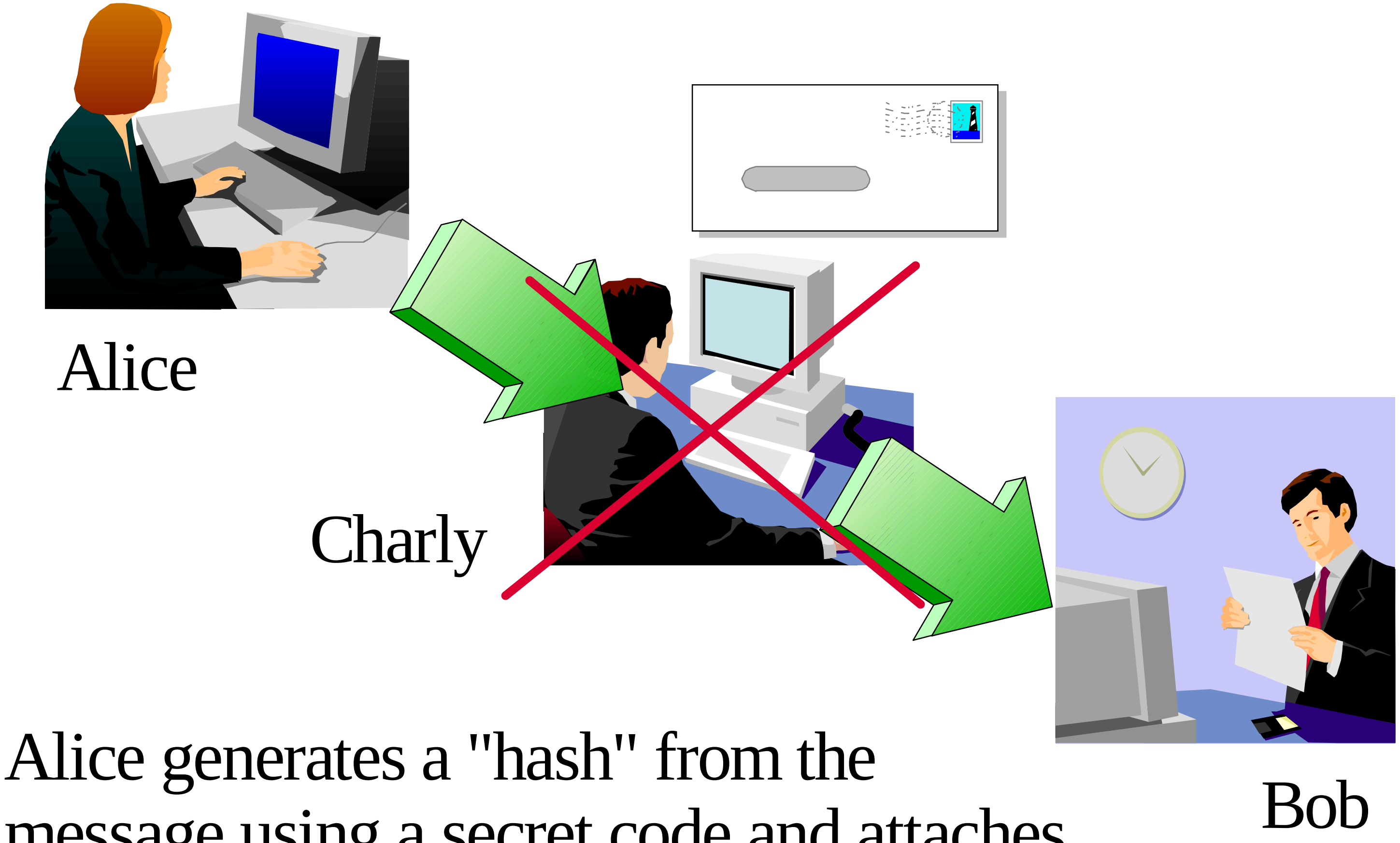
- Keeping secrets
  - Alice wants to send Bob confidential information,
  - Charly should not be able to read it.
- Proving identity
  - Bob receives a message from Alice. How he can be sure that it is really from Alice?
- Verifying information
  - Bob receives a message from Alice. How he can be sure that the content has not been modified?

Keeping Secrets



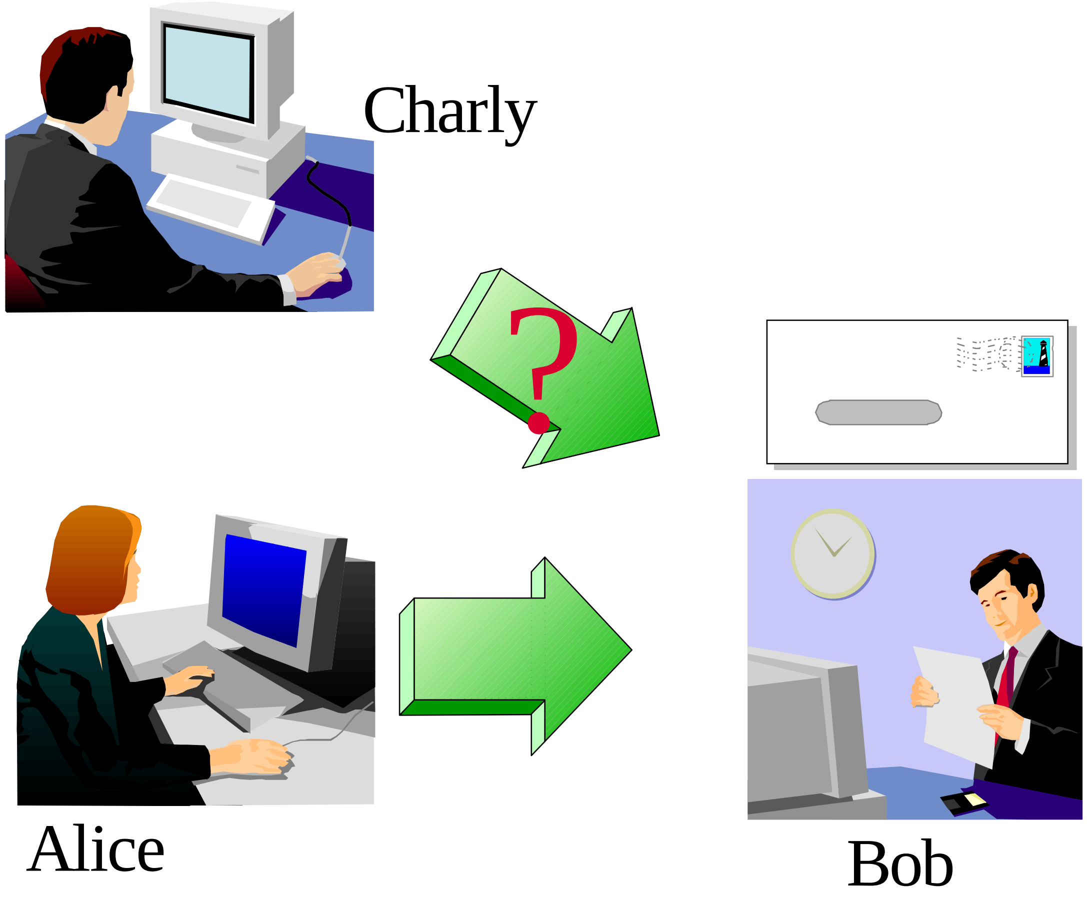
Alice encrypts the message with a secret code that only she and Bob knows

Verifying Information



Alice generates a "hash" from the message using a secret code and attaches it to the message. Bob also generates the hash from the received message and compares it.

Proving Identity



Alice "signs" the message by attaching a secret phrase that only she and Bob knows

Secret Key Cryptography (symmetric)

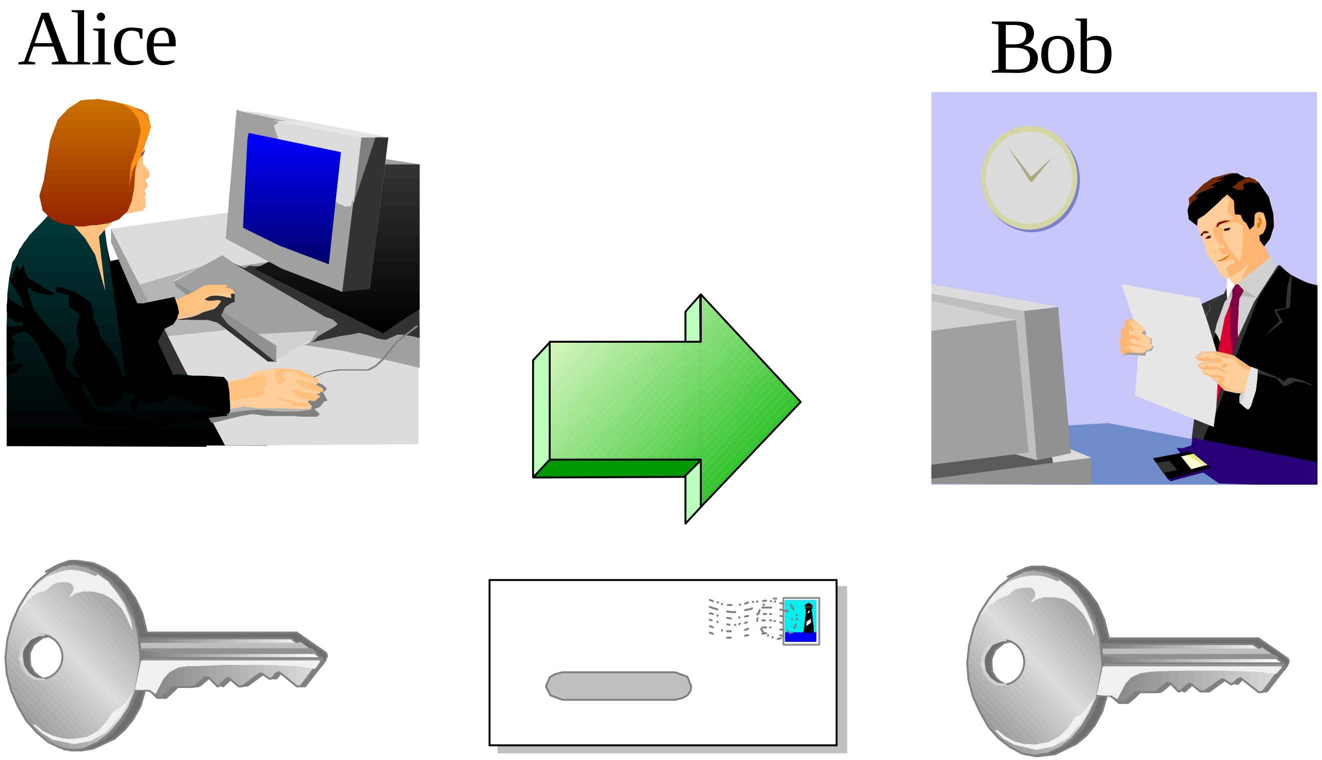
- Both parties know the same secret code (key)
- The key must be kept secret

- Encryption algorithm = mathematical transformation of the data with the key

– DES	Data Encryption standard
– 3DES	Triple strength DES
– AES	Advanced Encryption Standard

- Typical key length: 40, 56, 128 or 256 bit

Secret Key Cryptography - continued



Alice encrypts the message with the secret key and sends it to Bob. Bob decrypts the message with the secret key.

Public Key Cryptography (asymmetric)

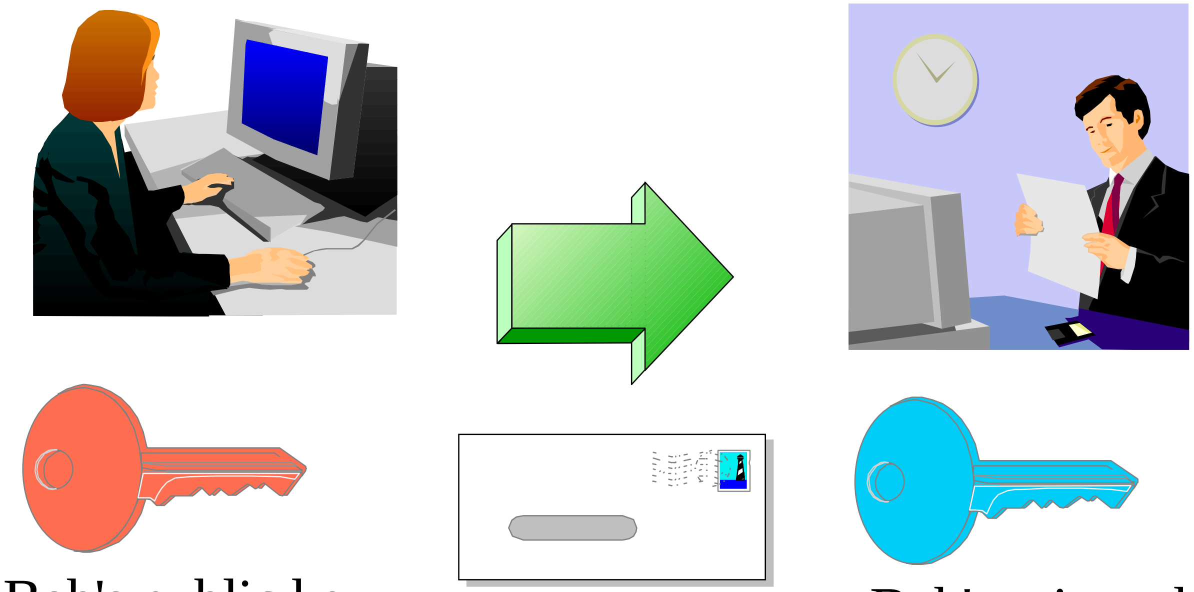
- One "public key" and one "private key“
- "Private key" is kept secret (private)
- "Public key" is published

- Asymmetric cryptography is based on mathematical problems, that are much easier to create than to solve

– RSA	Rivest Shamir Adleman
– DSA	Digital Signature Algorithm
– DHE	Diffie Hellman Algorithm

- Typical key length: 512, 1024 or 2048 bit

Public Key Cryptography - Encrypting



Alice

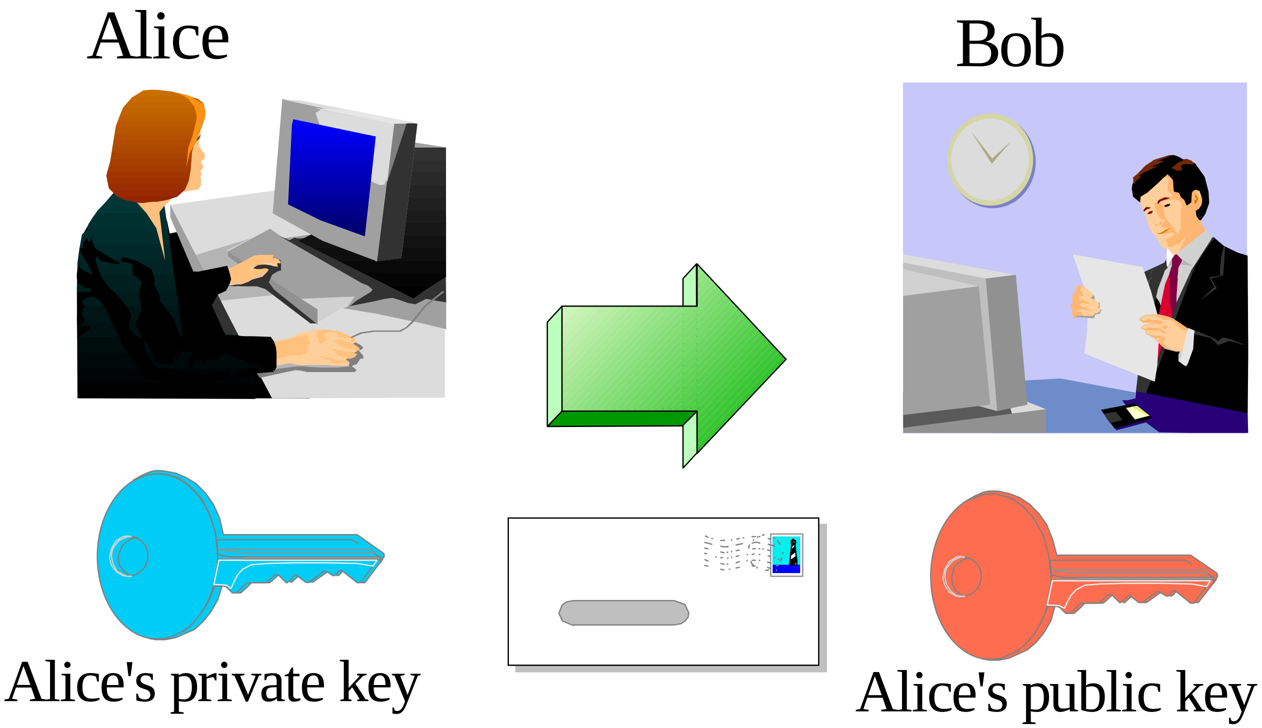
Bob

Bob's public key

Bob's private key

Alice encrypts the message using Bobs public key and sends it to Bob. Bob decrypts it using his private key.
Since only Bob knows his private key, only he can read the message.

Public Key Cryptography - Signing



Alice

Bob

Alice's private key

Alice's public key

Alice encrypts the message using her private key and sends it to Bob. Bob decrypts it using Alice's public key.
The message is "signed" by Alice since it can only be decrypted using **her** public key.

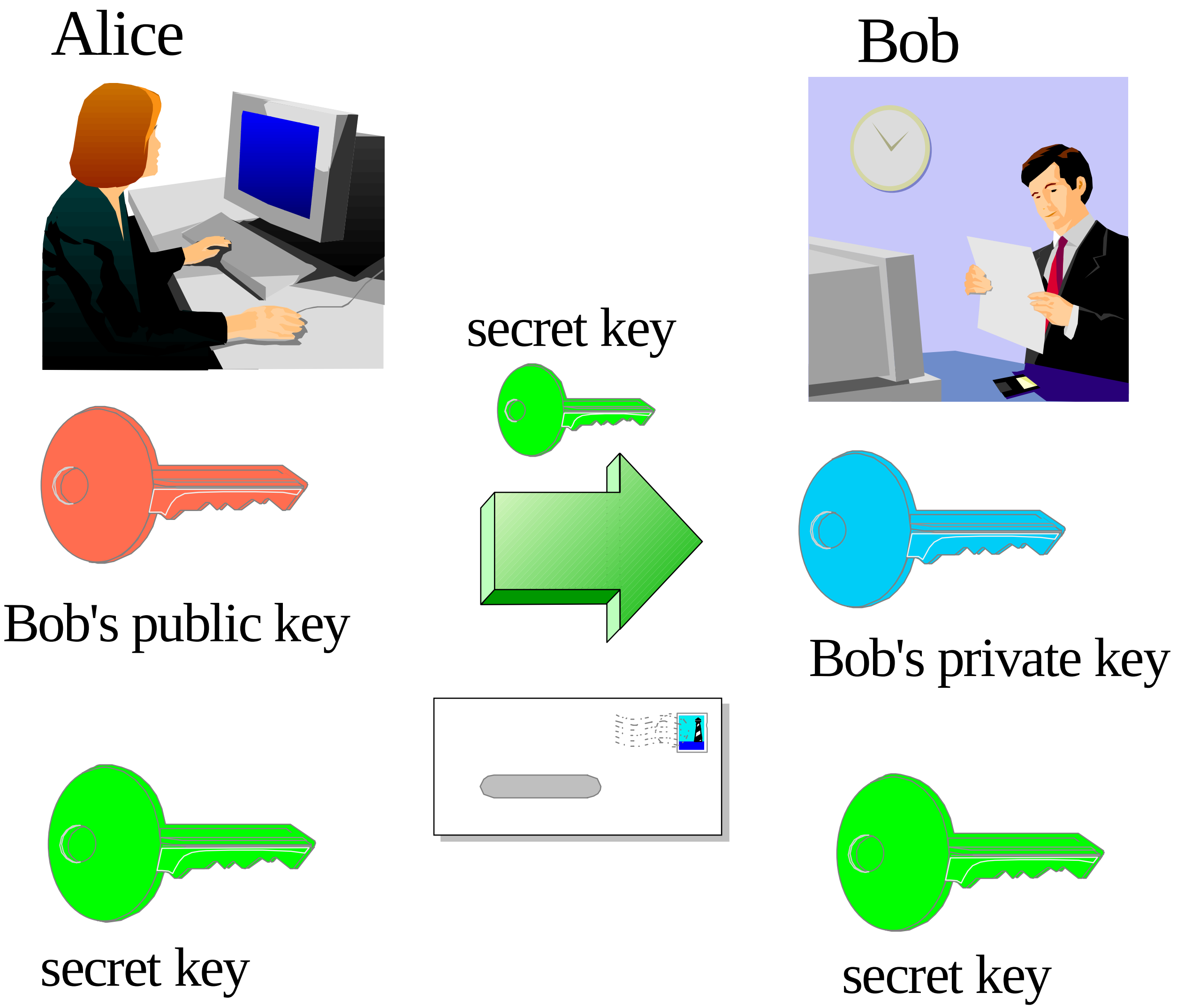
Combined Symmetric and Asymmetric Cryptography

Asymmetric cryptography is very CPU-time consuming

- Use asymmetric cryptography only for secret key exchange
- Data encryption uses symmetric cryptography
- Secret key is generated by random

- SSL also uses this mechanism

Combined Symmetric and Asymmetric Cryptography



Alice

Bob

secret key

Bob's public key

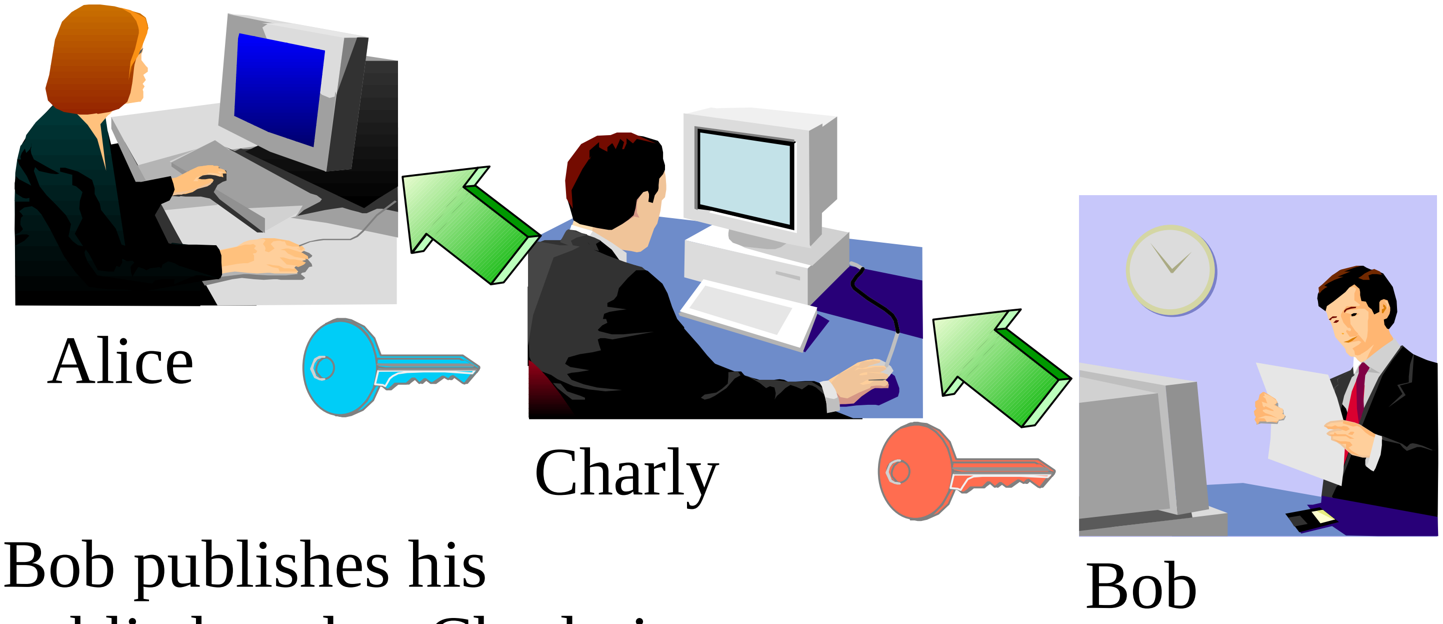
Bob's private key

secret key

secret key

Key Management

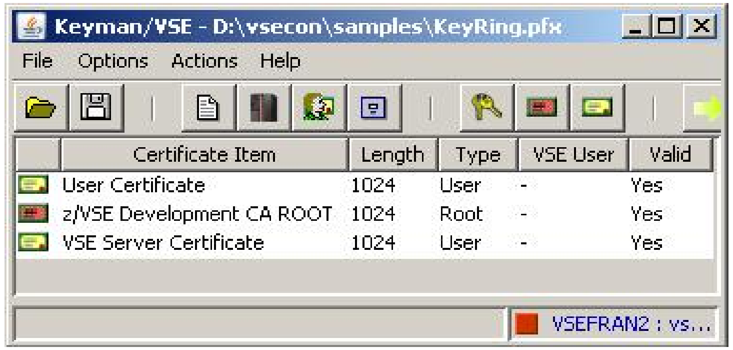
- Key exchange is not trivial:
  - Is the public key really from the right person?



Bob publishes his
public key, but Charly intercepts
this and instead sends his public key to Alice.

Key Management

- Key Management is not trivial
  - Key must often be kept secure for a very long time
  - You must be able to associate the encrypted data with the corresponding key(s)
  - Encrypted data and the corresponding key(s) must be strictly separated
- Keyman/VSE
  - Creation of RSA keys and digital certificates
  - Upload of keys and certificates to VSE
  - Creation of PKCS#12 keyring files (use with Java-based connector or import into a Web browser)
  - Download from VSE Homepage http://www.ibm.com/servers/eserver/zseries/zvse/downloads/#vkeyman



Certificate Item	Length	Type	VSE User	Valid
User Certificate	1024	User	-	Yes
z/VSE Development CA ROOT	1024	Root	-	Yes
VSE Server Certificate	1024	User	-	Yes

Certificates

- A certificate contains the following items
  - The subject (name of the person)
  - The subject's public key
  - Period of validity
  - The issuer
  - Issuers signature

- The issuer "signs" the certificate by encrypting a hash of the certificate content with his private key

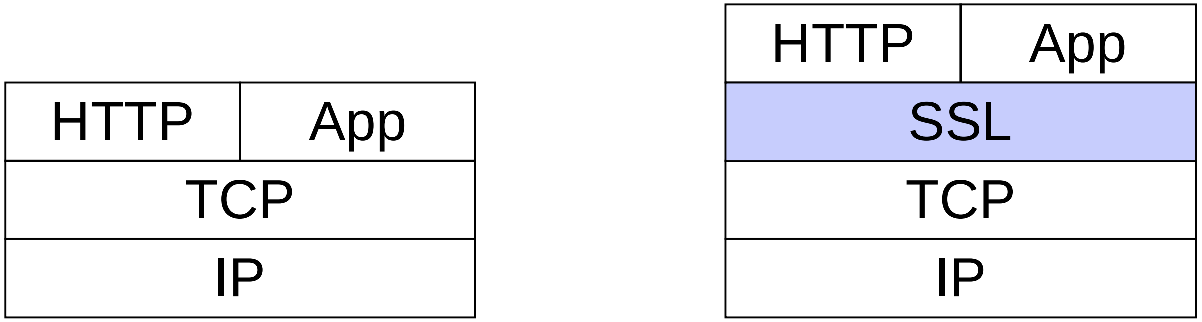
- Everyone can check the sign by decrypting it with the issuers public key

Certificate Authorities

- A certificate is issued by a certificate authority (CA)
- If a user trusts the certificate authority, he can trust the certificates issued by this CA
- CAs identify itself with a "self signed certificate":
  - The public key in the certificate is also the public key used to decrypt the signature
  - Subject and issuer are the same
- It is possible to build certificate hierarchies
- Certificate revocation lists are used to mark certificates that have been issued by error

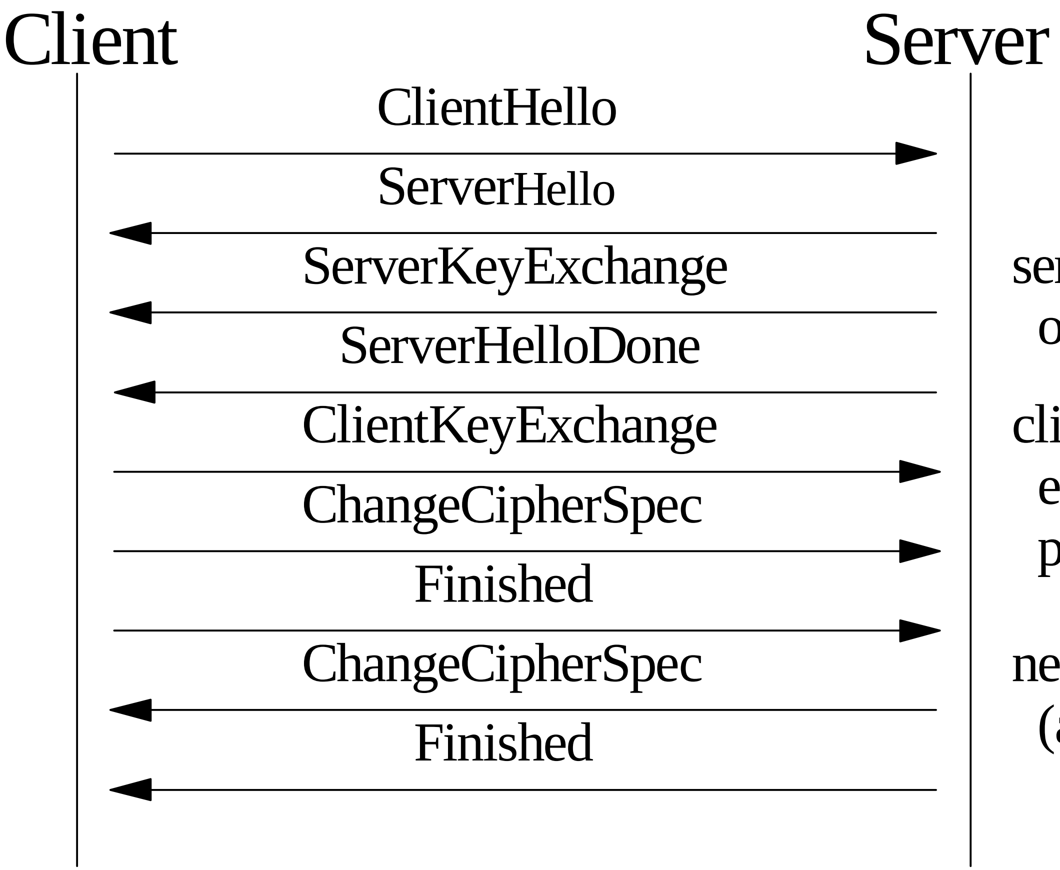
SSL (Secure Socket Layer)

- SSL provides a communication channel with message integrity, authentication, and confidentiality
- SSL is a widely used protocol
  - Secure HTTP (HTTPS) is used very often in the Internet
- SSL uses a TCP connection to transfer encrypted messages
  - Uses asymmetric cryptography for session initiating
  - Uses symmetric cryptography for data encryption
- As the name implies, SSL is a layer on top of TCP



SSL Protocol

- The SSL protocol defines a set of messages



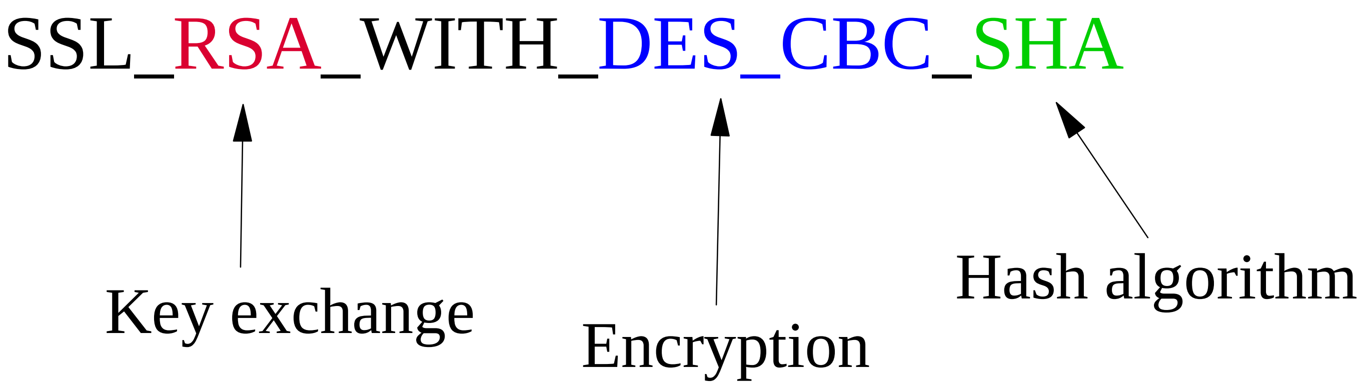
servers public key
or certificate

clients session key
encrypted with servers
public key

negoiation of options
(algorithms)

Cipher Suites

- Cipher suites defines the algorithms used:
  - For key exchange
  - For encryption
  - For hash algorithm



SSL_RSA_WITH_DES_CBC_SHA

Key exchange

Encryption

Hash algorithm

Session Caching

- "SSL Session" means
  - Secret key used for data encryption
  - Negotiated algorithms
- Establishing a SSL Session is a complex and time consuming mechanism
- Session caching allows to reuse previously negotiated SSL parameters
- No need of repeating the negotiations or authentications
  - The same symmetric key is used
- The connection becomes more unsecured
- A SSL Session time-out defines how long a session is kept alive

SSL for VSE

- SSL for VSE is part of the TCP/IP for VSE base
  - Enabled with the Application Pak
  - Integrated into TCP/IP for VSE
- Supports SSL 3.0 and TLS 1.0
  - Key exchange: RSA
  - Data Encryption: DES and Triple DES, AES
  - Hash algorithm: MD5, SHA
  - Supports X.509v3 PKI Certificates
- SSL daemon implementation for HTTPS, Telnet
- SSL API compatible with the OS/390 SSL API
- Uses Hardware Crypto acceleration if available

SSL Daemon (SSLD)

- Define a SSL daemon for each TCP port that you want to secure:

```
DEFINE TLSD,ID=MYSSLD,
             PORT=443,
             HTTPS port
             PASSPORT=443,
             CIPHER=0A096208,       Cipher suites
             CERTLIB=CRYPTO,        library name
             CERTSUB=KEYRING,       sub library name
             CERTMEM=MYKEY,         member name
             TYPE=1,                server application
             MINVERS=0300,          SSL 3.0
             DRIVER=SSLD            Driver phase name
```

Secure Socket Layer API

- Compatible to OS/390 SSL API
- Functions available for
  - Session initiating
  - Sending/receiving data
  - Ending a session
- SSL API is based on Socket API
- SSL API can be called from
  - LE-C programs
  - Assembler programs

Secure Socket Layer - Concepts

- When using SSL, you need to have a set of certificates and keys
  - A Public/Private key pair
  - Root Certificate
    - Certificate of a Certificate Authority (CA) that has issued the other certificates
  - Your own certificate
    - A certificate that was issued to you by a certificate authority
  - Partner Certificate(s)
    - Certificate(s) of your communication partners

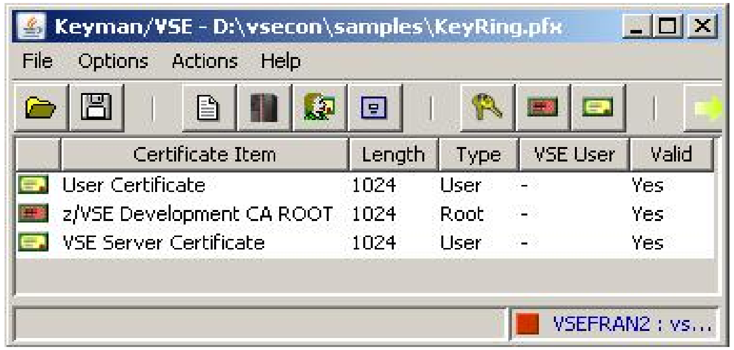
- When you do HTTPS with your browser usually already contains these keys and certificates

Secure Socket Layer - Concepts

- For production purposes, certificates are usually issued by a well known and trusted Certificate Authorities (CA)
  - For example Thawte, VeriSign
  - Usually this cost money
- For in-house use (Intranet), you can have your own Company-wide Certificate Authority
  - Certificates are trusted inside your company, but not outside
- For test purposes you can use self-signed Certificates (you are your own Certificate Authority)
  - Nobody trusts these Certificates (except you)

Secure Socket Layer - Setup

- To setup all required keys and certificates, it is recommended to use the Tool Keyman/VSE
  - Download from VSE Homepage
    http://www.ibm.com/servers/eserver/zseries/zvse/downloads/#vkeyman
- Supports creation of keys and CA-signed or self-signed Certificates for use with SSL



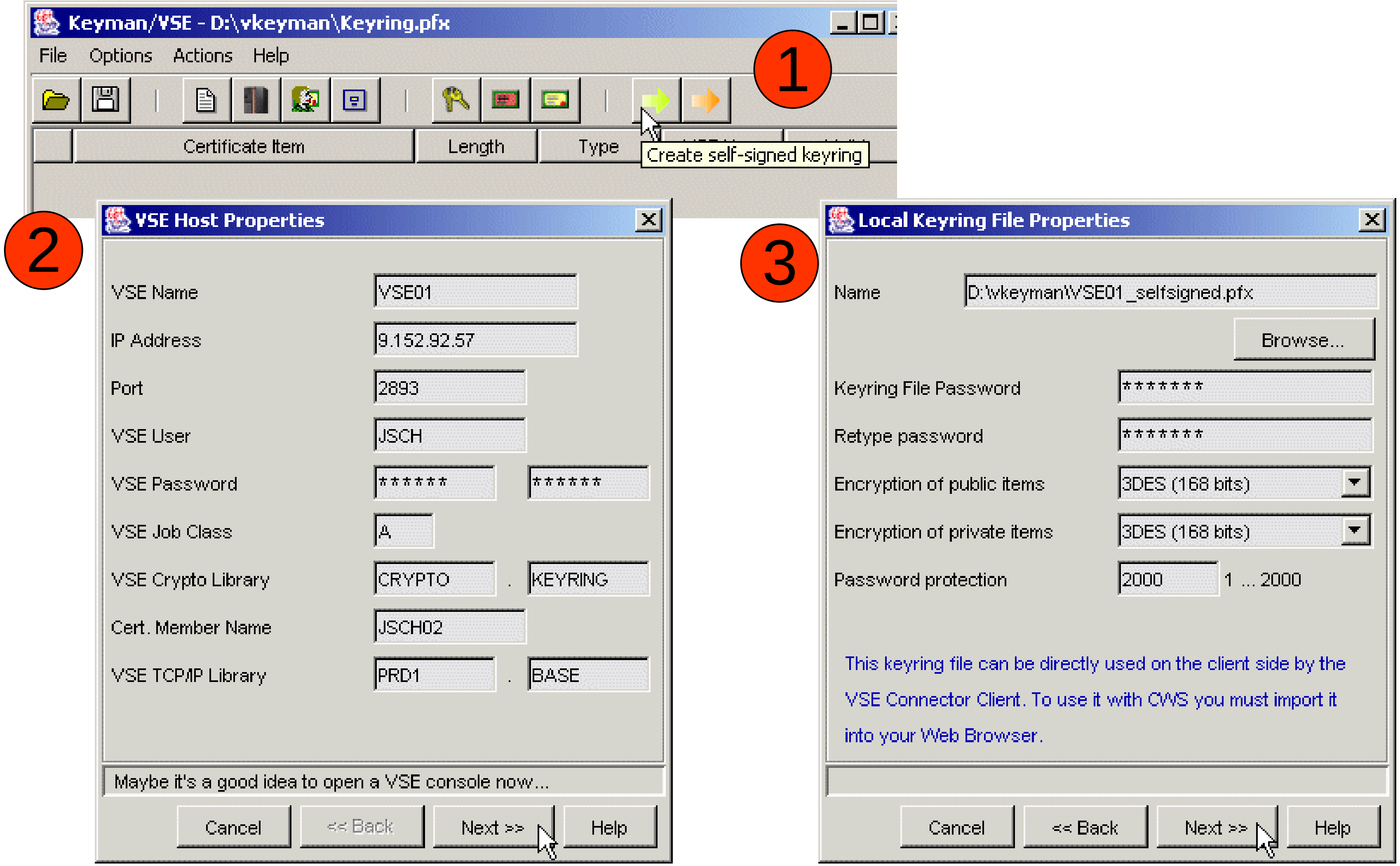
	Certificate Item	Length	Type	VSE User	Valid
🟨	User Certificate	1024	User	-	Yes
🟥	z/VSE Development CA ROOT	1024	Root	-	Yes
🟨	VSE Server Certificate	1024	User	-	Yes

- Online documentation contains ‘How to’ sections with step by step descriptions for creating keys and certificates

Setup a self signed certificate

- Steps for creating a self-signed certificate:
  1. Create an RSA key pair
  2. Create a self-signed root certificate
  3. Create a VSE server certificate
  4. Sign the request with your root certificate
  5. Make your VSE host ready for uploading
  6. Upload the key to VSE
  7. Upload the root certificate to VSE
  8. Upload the server certificate to VSE
  9. Save your local keyring file
- Use Wizard Dialog “Create self-signed keyring”

Setup a self signed certificate - Wizard



1

Keyman/VSE - D:\vkeyman\Keyring.pfx

2

VSE Host Properties

VSE Name	VSE01	
IP Address	9.152.92.57	
Port	2893	
VSE User	JSCH	
VSE Password	******	******
VSE Job Class	A	
VSE Crypto Library	CRYPTO .	KEYRING
Cert. Member Name	JSCH02	
VSE TCP/IP Library	PRD1 .	BASE

Maybe it's a good idea to open a VSE console now...

3

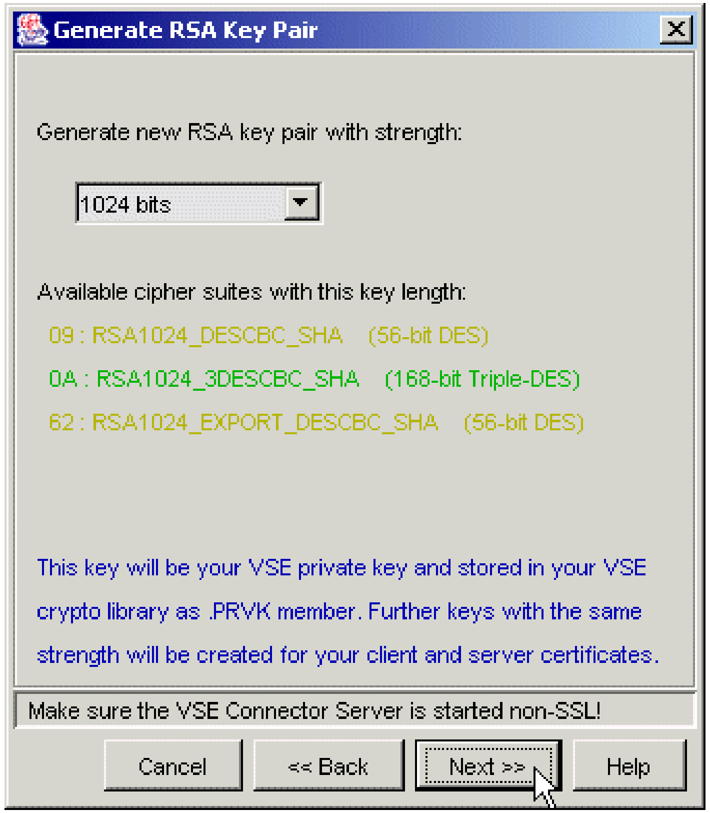
Local Keyring File Properties

Name	D:\vkeyman\VSE01_selfsigned.pfx
Keyring File Password	*******
Retype password	*******
Encryption of public items	3DES (168 bits)
Encryption of private items	3DES (168 bits)
Password protection	2000 1 ... 2000

This keyring file can be directly used on the client side by the VSE Connector Client. To use it with CWS you must import it into your Web Browser.

Setup a self signed certificate - Wizard

4



Generate RSA Key Pair

Generate new RSA key pair with strength:

1024 bits

Available cipher suites with this key length:

09 : RSA1024_DESCBC_SHA (56-bit DES)

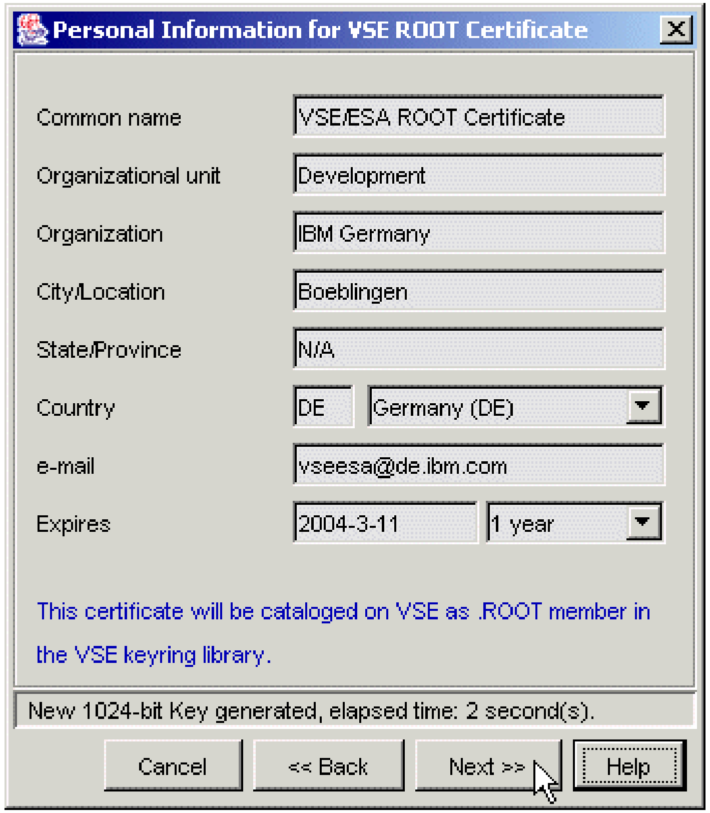
0A : RSA1024_3DESCBC_SHA (168-bit Triple-DES)

62 : RSA1024_EXPORT_DESCBC_SHA (56-bit DES)

This key will be your VSE private key and stored in your VSE crypto library as .PRVK member. Further keys with the same strength will be created for your client and server certificates.

Make sure the VSE Connector Server is started non-SSL!

5



Personal Information for VSE ROOT Certificate

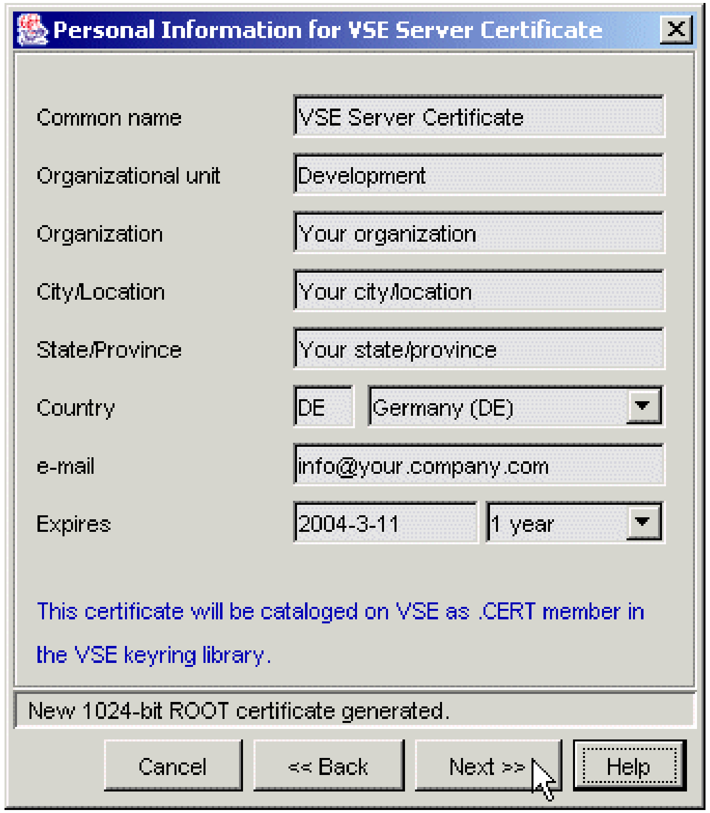
Common name	VSE/ESA ROOT Certificate
Organizational unit	Development
Organization	IBM Germany
City/Location	Boeblingen
State/Province	N/A
Country	DE Germany (DE)
e-mail	vseesa@de.ibm.com
Expires	2004-3-11 1 year

This certificate will be cataloged on VSE as .ROOT member in the VSE keyring library.

New 1024-bit Key generated, elapsed time: 2 second(s).

Setup a self signed certificate - Wizard

6



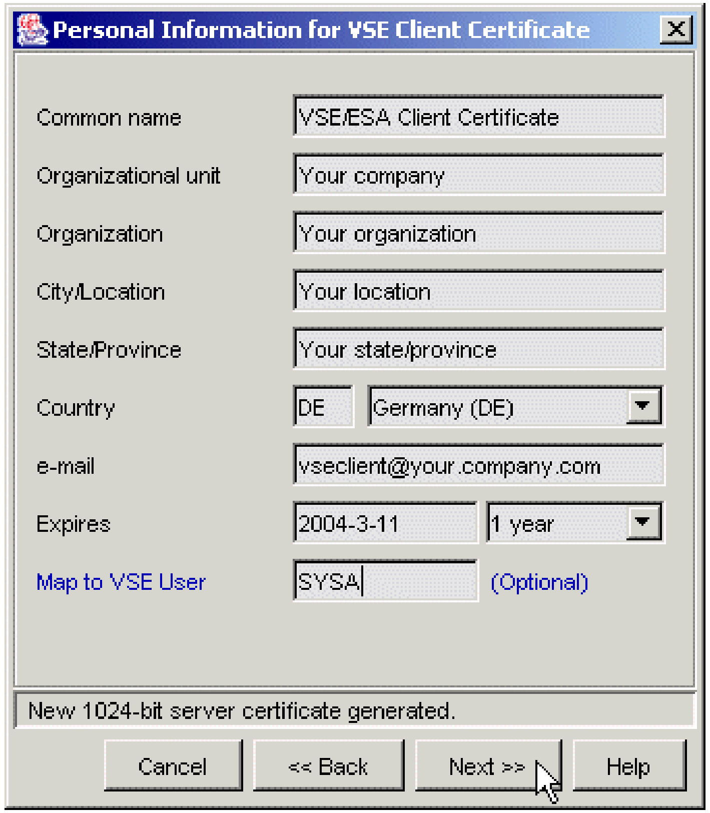
Personal Information for VSE Server Certificate

Common name	VSE Server Certificate
Organizational unit	Development
Organization	Your organization
City/Location	Your city/location
State/Province	Your state/province
Country	DE Germany (DE)
e-mail	info@your.company.com
Expires	2004-3-11 1 year

This certificate will be cataloged on VSE as .CERT member in the VSE keyring library.

New 1024-bit ROOT certificate generated.

7



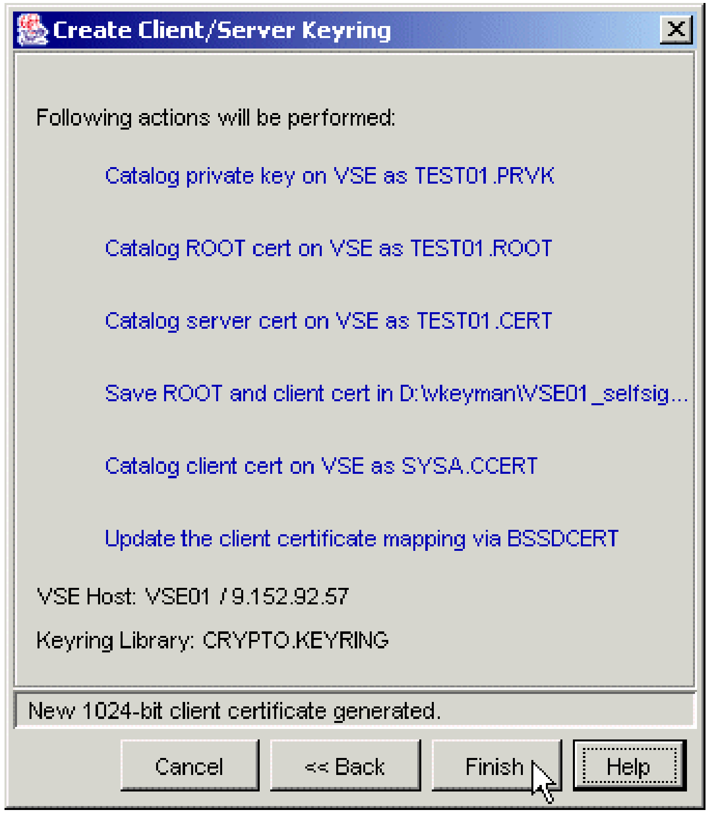
Personal Information for VSE Client Certificate

Common name	VSE/ESA Client Certificate
Organizational unit	Your company
Organization	Your organization
City/Location	Your location
State/Province	Your state/province
Country	DE Germany (DE)
e-mail	vseclient@your.company.com
Expires	2004-3-11 1 year
Map to VSE User	SYSA (Optional)

New 1024-bit server certificate generated.

Setup a self signed certificate - Wizard

8



Create Client/Server Keyring

Following actions will be performed:

Catalog private key on VSE as TEST01.PRVK

Catalog ROOT cert on VSE as TEST01.ROOT

Catalog server cert on VSE as TEST01.CERT

Save ROOT and client cert in D:\keyman\VSE01_selfsig...

Catalog client cert on VSE as SYSA.CCERT

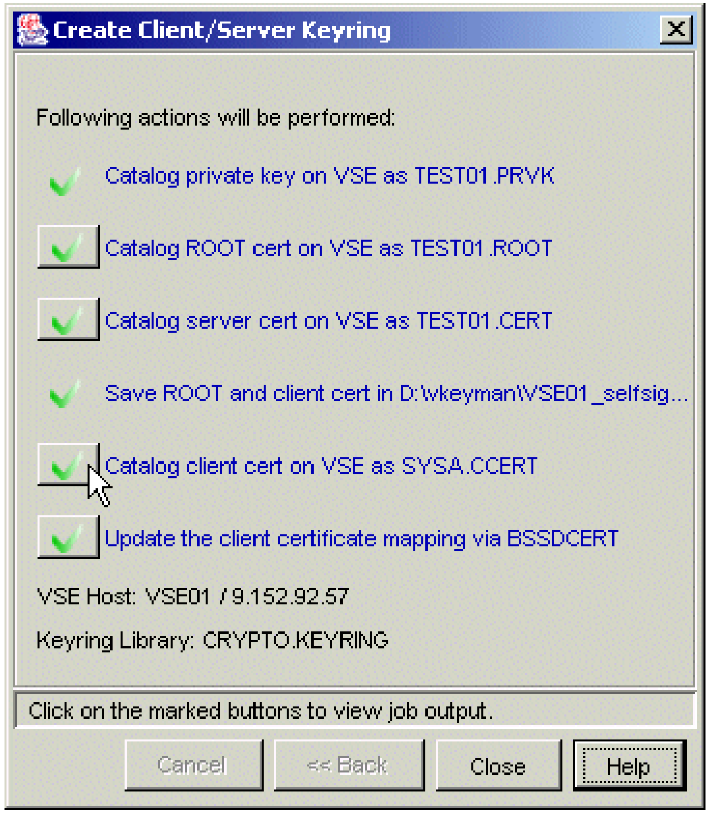
Update the client certificate mapping via BSSDCERT

VSE Host: VSE01 / 9.152.92.57

Keyring Library: CRYPTO.KEYRING

New 1024-bit client certificate generated.

9



Create Client/Server Keyring

Following actions will be performed:

✓ Catalog private key on VSE as TEST01.PRVK

✓ Catalog ROOT cert on VSE as TEST01.ROOT

✓ Catalog server cert on VSE as TEST01.CERT

✓ Save ROOT and client cert in D:\keyman\VSE01_selfsig...

✓ Catalog client cert on VSE as SYSA.CCERT

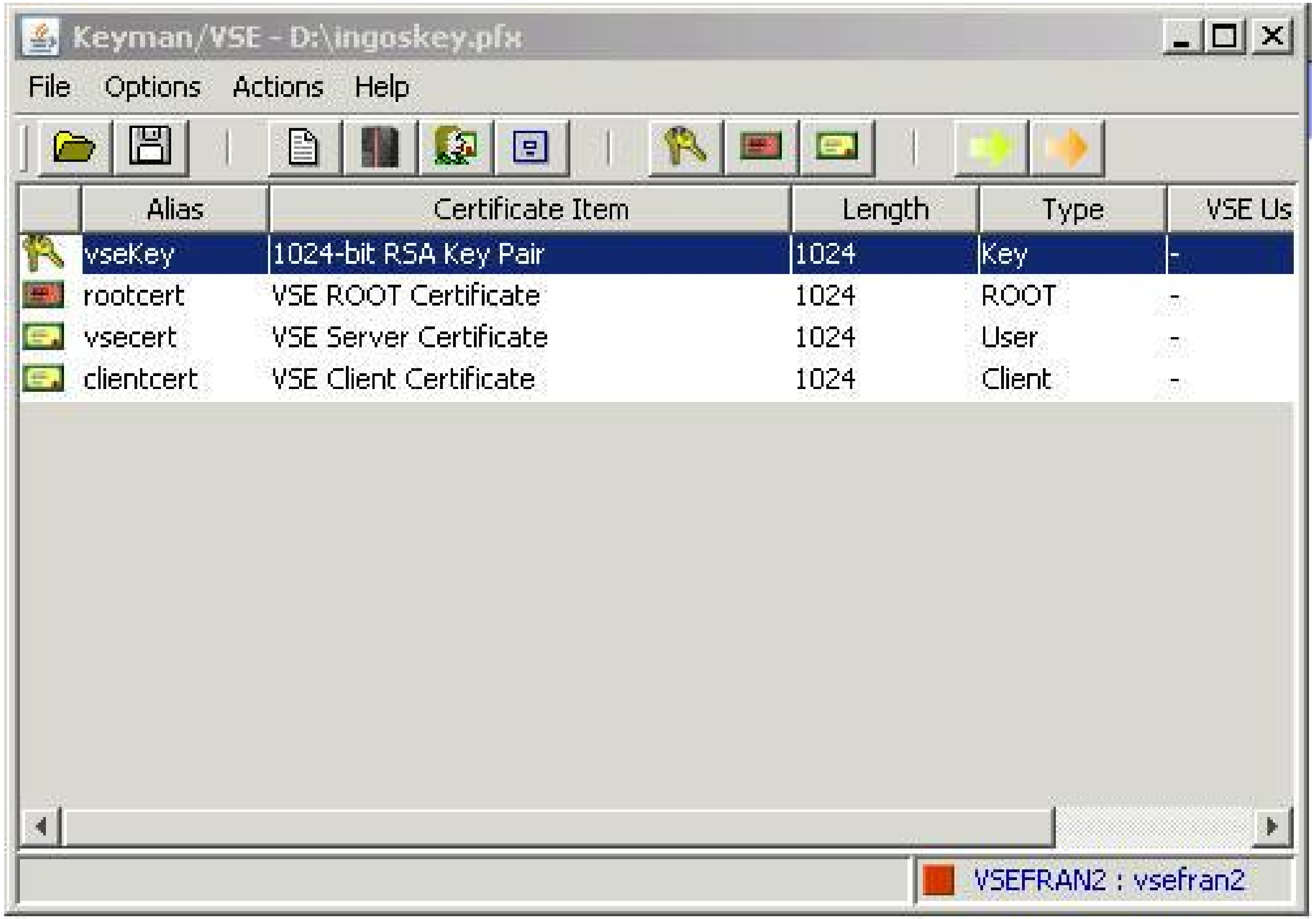
✓ Update the client certificate mapping via BSSDCERT

VSE Host: VSE01 / 9.152.92.57

Keyring Library: CRYPTO.KEYRING

Click on the marked buttons to view job output.

Setup a self signed certificate - Wizard

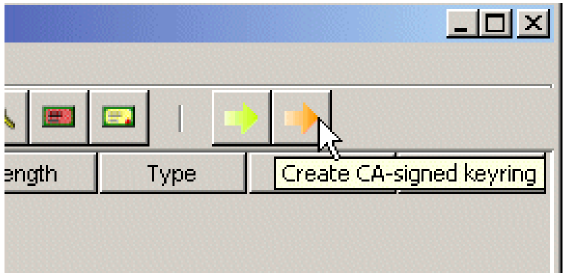


	Alias	Certificate Item	Length	Type	VSE Us
	vseKey	1024-bit RSA Key Pair	1024	Key	-
	rootcert	VSE ROOT Certificate	1024	ROOT	-
	vsecert	VSE Server Certificate	1024	User	-
	clientcert	VSE Client Certificate	1024	Client	-

Setup a CA signed certificate

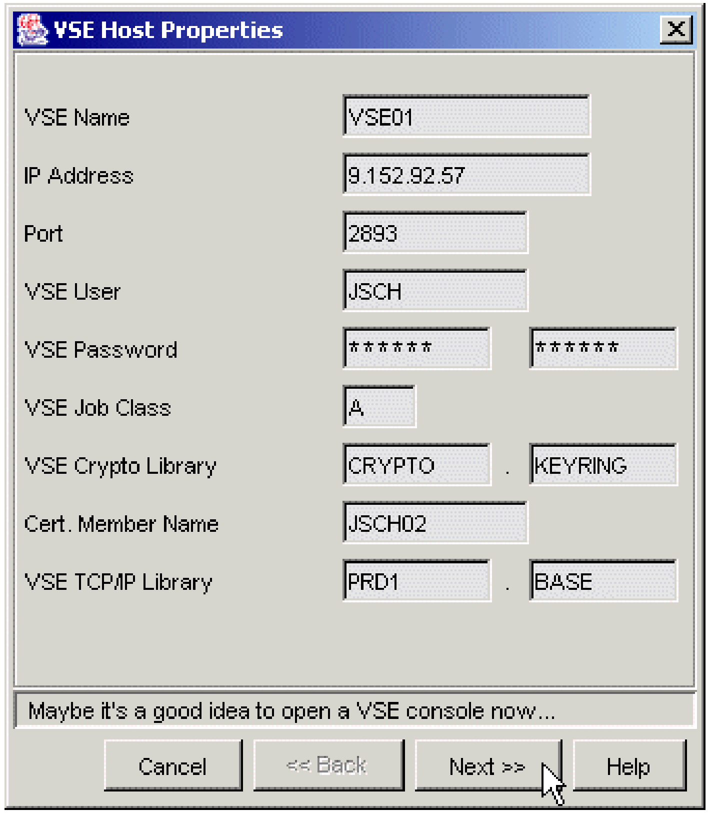
- Steps for creating a CA signed certificate:
  1. Create an RSA key pair
  2. Create a certificate request
  3. Copy request to clipboard
  4. Go to the CA's web site (e.g. Thawte, VeriSign)
  5. Request the server certificate on the CA’s web site
  6. Import signed server cert into Keyman/VSE
  7. Get the CA's public root certificate
  8. Make your VSE host ready for uploading
  9. Upload the key to VSE
  10. Upload the root certificate to VSE
  11. Upload the server certificate to VSE
  12. Save your local keyring file
- Use Wizard Dialog “Create CA signed keyring”

Setup a CA signed certificate - Wizard



1

2

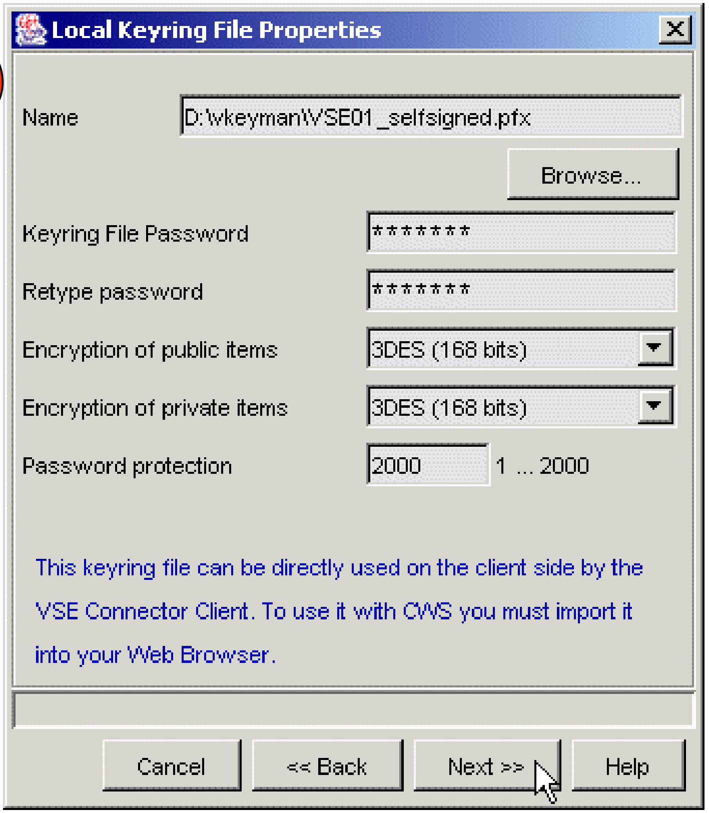


VSE Host Properties

VSE Name	VSE01
IP Address	9.152.92.57
Port	2893
VSE User	JSCH
VSE Password	****** ******
VSE Job Class	A
VSE Crypto Library	CRYPTO . KEYRING
Cert. Member Name	JSCH02
VSE TCP/IP Library	PRD1 . BASE

Maybe it's a good idea to open a VSE console now...

3



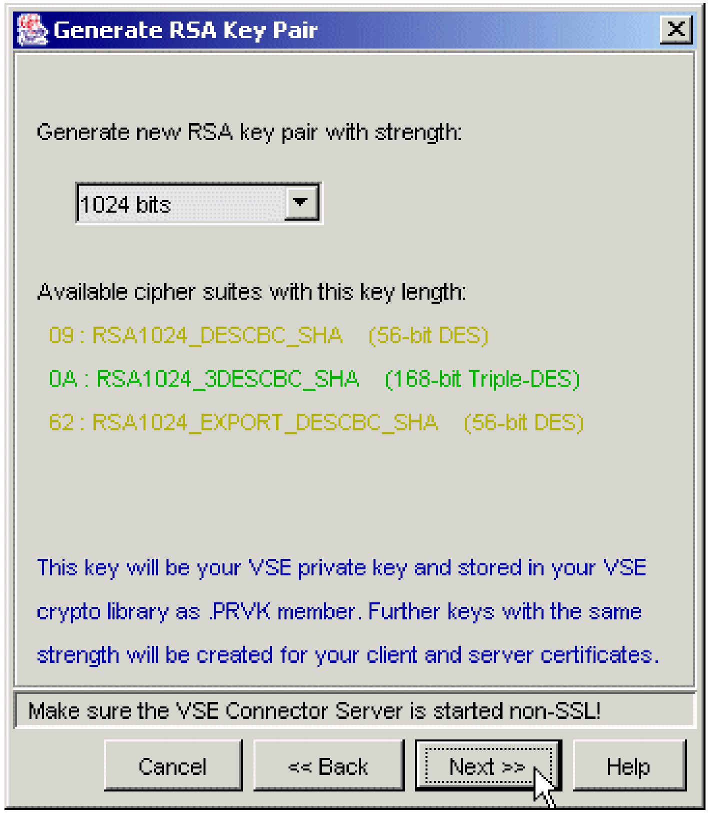
Local Keyring File Properties

Name	D:\wkeyman\VSE01_selfsigned.pfx
Keyring File Password	*******
Retype password	*******
Encryption of public items	3DES (168 bits)
Encryption of private items	3DES (168 bits)
Password protection	2000 1 ... 2000

This keyring file can be directly used on the client side by the VSE Connector Client. To use it with CWS you must import it into your Web Browser.

Setup a CA signed certificate - Wizard

4



Generate RSA Key Pair

Generate new RSA key pair with strength:

1024 bits

Available cipher suites with this key length:

09 : RSA1024_DESCBC_SHA (56-bit DES)

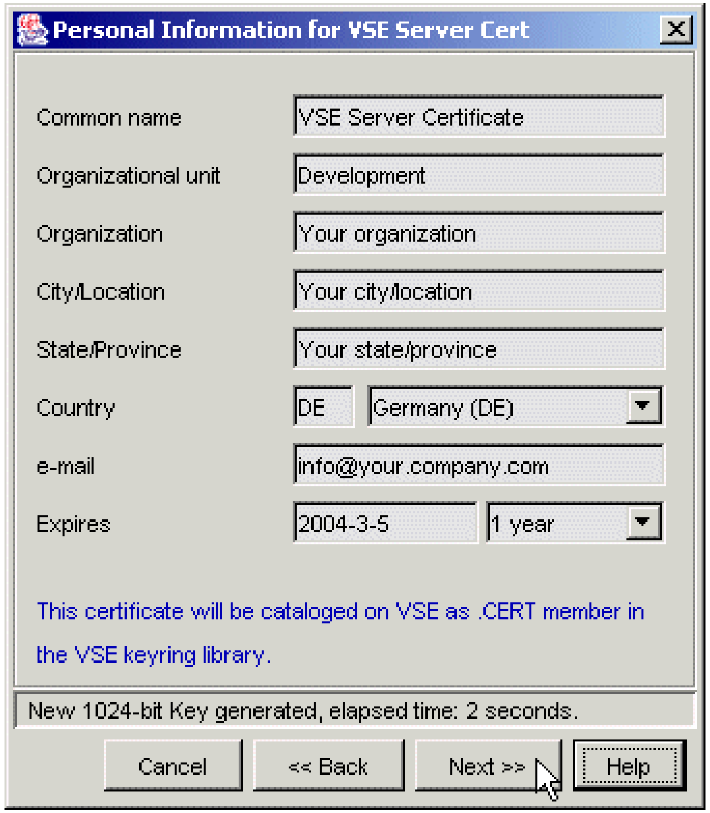
0A : RSA1024_3DESCBC_SHA (168-bit Triple-DES)

62 : RSA1024_EXPORT_DESCBC_SHA (56-bit DES)

This key will be your VSE private key and stored in your VSE crypto library as .PRVK member. Further keys with the same strength will be created for your client and server certificates.

Make sure the VSE Connector Server is started non-SSL!

5



Personal Information for VSE Server Cert

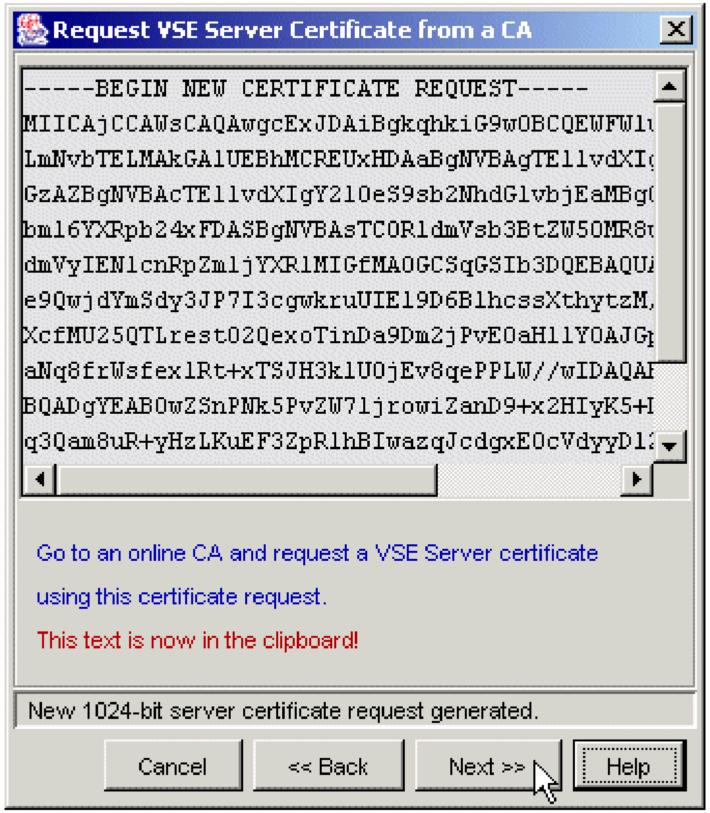
Common name	VSE Server Certificate
Organizational unit	Development
Organization	Your organization
City/Location	Your city/location
State/Province	Your state/province
Country	DE Germany (DE)
e-mail	info@your.company.com
Expires	2004-3-5 1 year

This certificate will be cataloged on VSE as .CERT member in the VSE keyring library.

New 1024-bit Key generated, elapsed time: 2 seconds.

Setup a CA signed certificate - Wizard

6



Request VSE Server Certificate from a CA

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIICAjCCAWsCAQAwgcExJDAiBgkqhkiG9w0BCQEWFWl
LmNvbTELMAkGA1UEBhMCREUxHDAaBgNVBAgTEllvdXI
GzAZBgNVBAcTEllvdXIgY2l0eS9sb2NhdGlvbjEaMBg
bml6YXRpb24xFDASBgNVBAsTC0RldmVsb3BtZW50MR8
dmVyIENlcnRpZmljYXRlMIGfMA0GCSqGSIb3DQEBAQU
e9QwjdYmSdy3JP7I3cgwkruUIEl9D6BlhcssXthytzM
XcfMU25QTLrest02QexoTinDa9Dm2jPvE0aHllY0AJG
aNq8frWsfexlRt+xTSJH3klU0jEv8qePPLW//wIDAQA
BQADgYEAB0wZSnPNk5PvZW7ljrowiZanD9+x2HIyK5+
q3Qam8uR+yHzLKuEF3ZpRlhBIwazqJcdgxE0cVdyyDl
```

Go to an online CA and request a VSE Server certificate using this certificate request.

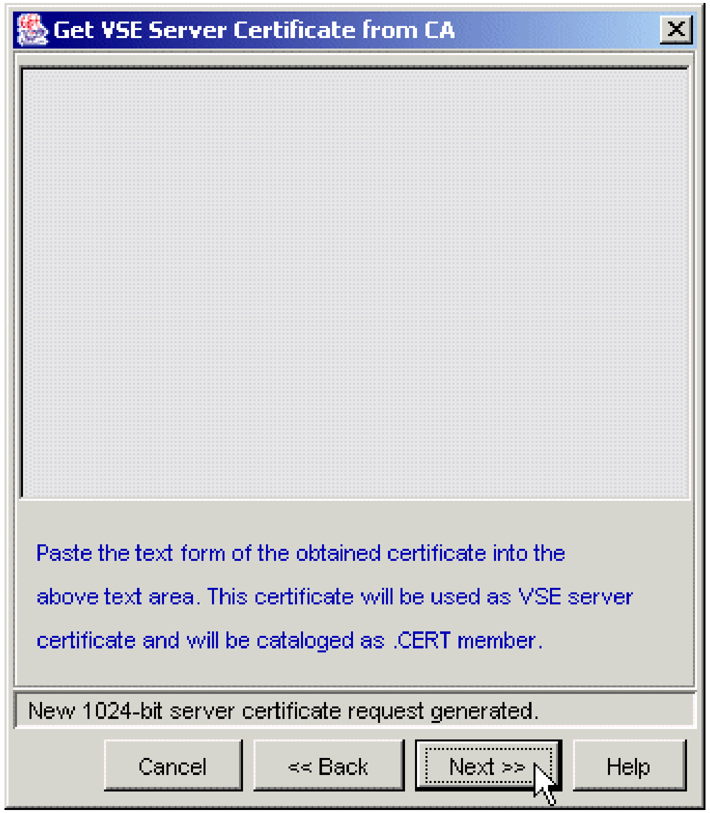
This text is now in the clipboard!

New 1024-bit server certificate request generated.

Copy and Paste the request into a CA's web site and let them sign the request.

You can paste the generated certificate into the text area on the following dialog box.

7

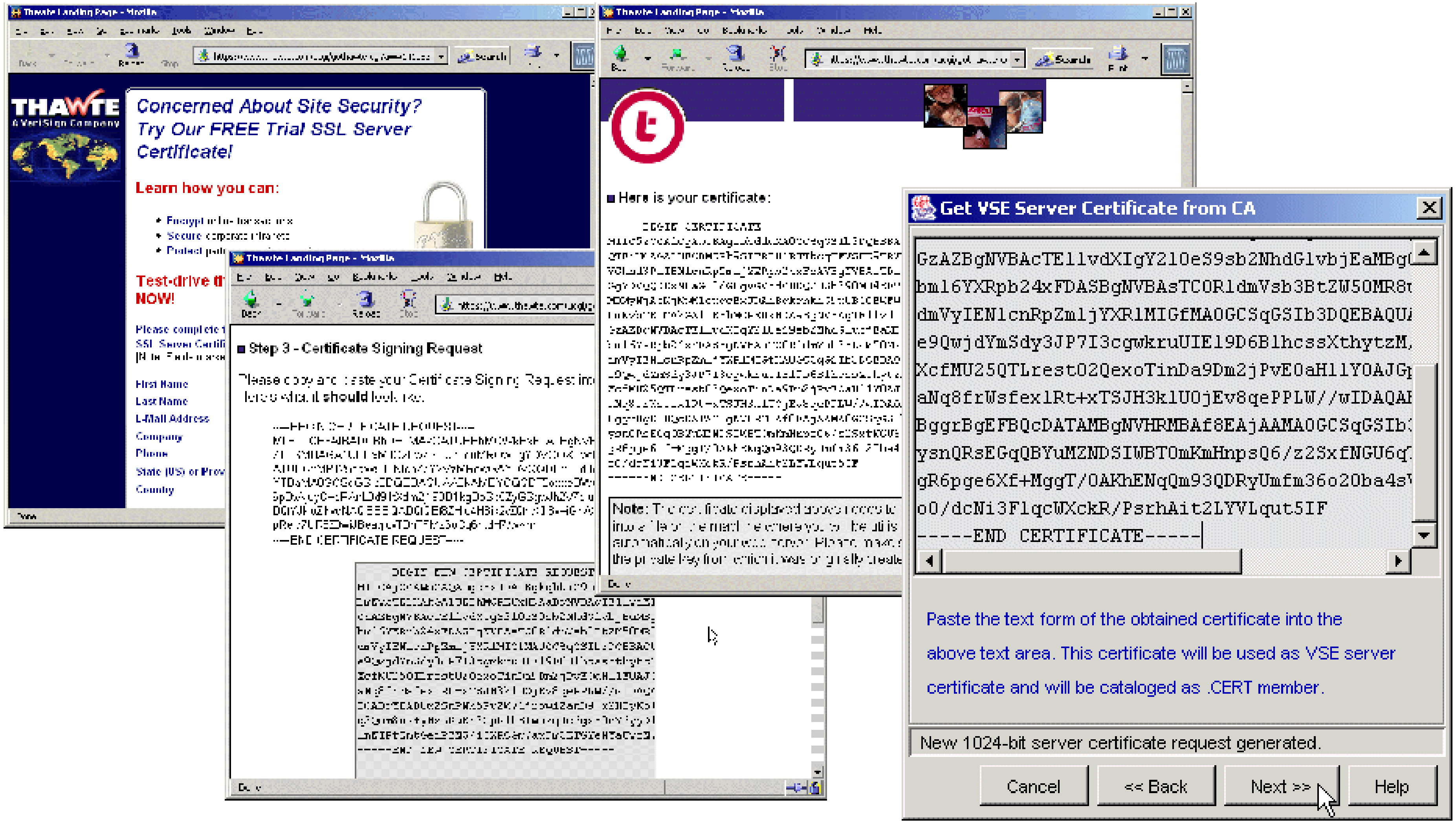


Get VSE Server Certificate from CA

Paste the text form of the obtained certificate into the above text area. This certificate will be used as VSE server certificate and will be cataloged as .CERT member.

New 1024-bit server certificate request generated.

Setup a CA signed certificate - Wizard



Thawte Landing Page - Mozilla

THAWTE
A VeriSign Company

Concerned About Site Security?
Try Our FREE Trial SSL Server Certificate!

Learn how you can:

- Encrypt [illegible]
- Secure corporate [illegible]
- Protect [illegible]

Test-drive [illegible] NOW!

Please complete [illegible]
SSL Server Certifi[illegible]

First Name
Last Name
E-Mail Address
Company
Phone
State (US) or Prov[illegible]
Country

Thawte Landing Page - Mozilla

■ Step 3 - Certificate Signing Request

Please copy and paste your Certificate Signing Request in[illegible] here is what it should look like:

```
-----BEGIN CERTIFICATE REQUEST-----
[illegible]
-----END CERTIFICATE REQUEST-----
```

```
-----BEGIN NEW CERTIFICATE REQUEST-----
[illegible]
-----END NEW CERTIFICATE REQUEST-----
```

Thawte Landing Page - Mozilla

■ Here is your certificate:

```
-----BEGIN CERTIFICATE-----
[illegible]
-----END CERTIFICATE-----
```

Note: The certificate displayed above [illegible] the private key from which it was originally create[illegible]

Get VSE Server Certificate from CA

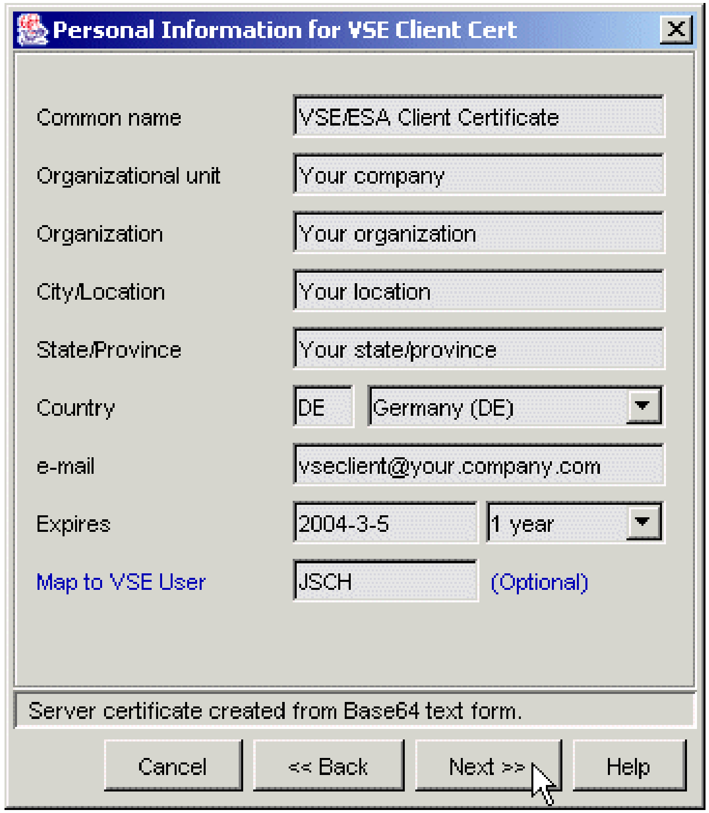
```
GzAZBgNVBAcTEllvdXIgY2l0eS9sb2NhdGlvbjEaMBg
bml6YXRpb24xFDASBgNVBAsTC0RldmVsb3BtZW50MR8
dmVyIEN1cnRpZmljYXRlMIGfMA0GCSqGSIb3DQEBAQU
e9QwjdYmSdy3JP7I3cgwkruUIEl9D6BlhcssXthytzM,
XcfMU25QTLrest02QexoTinDa9Dm2jPvE0aHllY0AJG
aNq8frWsfexlRt+xTSJH3klU0jEv8qePPLW//wIDAQA
BggrBgEFBQcDATAMBgNVHRMBAf8EAjAAMA0GCSqGSIb
ysnQRsEGqQBYuMZNDSIWBTOmKmHnpsQ6/z2SxfNGU6q
gR6pge6Xf+MggT/0AKhENqQm93QDRyUmfm36o20ba4s
o0/dcNi3FlqcWXckR/PsrhAit2LYVLqut5IF
-----END CERTIFICATE-----
```

Paste the text form of the obtained certificate into the above text area. This certificate will be used as VSE server certificate and will be cataloged as .CERT member.

New 1024-bit server certificate request generated.

Setup a CA signed certificate - Wizard

8

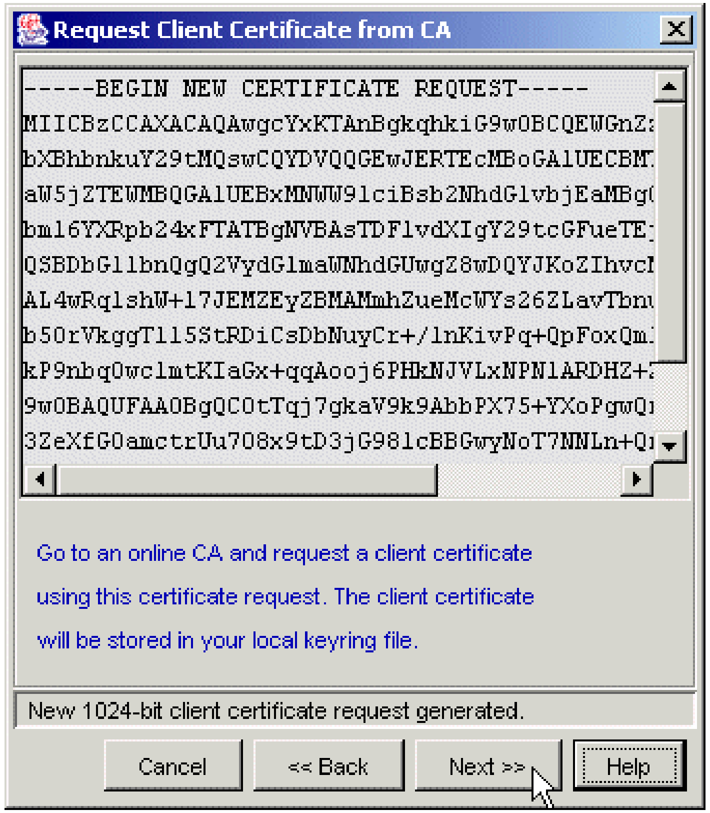


Personal Information for VSE Client Cert

Common name	VSE/ESA Client Certificate	
Organizational unit	Your company	
Organization	Your organization	
City/Location	Your location	
State/Province	Your state/province	
Country	DE	Germany (DE)
e-mail	vseclient@your.company.com	
Expires	2004-3-5	1 year
Map to VSE User	JSCH	(Optional)

Server certificate created from Base64 text form.

9



Request Client Certificate from CA

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIICBzCCAXACAQAwgcYxKTAnBgkqhkiG9w0BCQEWGnZ
bXBhbnkuY29tMQswCQYDVQQGEwJERTEcMBoGA1UECBM
aW5jZTEWMBQGA1UEBxMNWW91ciBsb2NhdGlvbjEaMBg
bml6YXRpb24xFTATBgNVBAsTDFlvdXIgY29tcGFueTE
QSBDbGllbnQgQ2VydGlmaWNhdGUwgZ8wDQYJKoZIhvc
AL4wRqlshW+17JEMZEyZBMAMmhZueMcWYs26ZLavTbn
b50rVkggTll5StRDiCsDbNuyCr+/lnKivPq+QpFoxQm
kP9nbq0wclmtKIaGx+qqAooj6PHkNJVLxNPNlARDHZ+
9w0BAQUFAA0BgQC0tTqj7gkaV9k9AbbPX75+YXoPgwQ
3ZeXfG0amctrUu708x9tD3jG98lcBBGwyNoT7NNLn+Q
```

Go to an online CA and request a client certificate using this certificate request. The client certificate will be stored in your local keyring file.

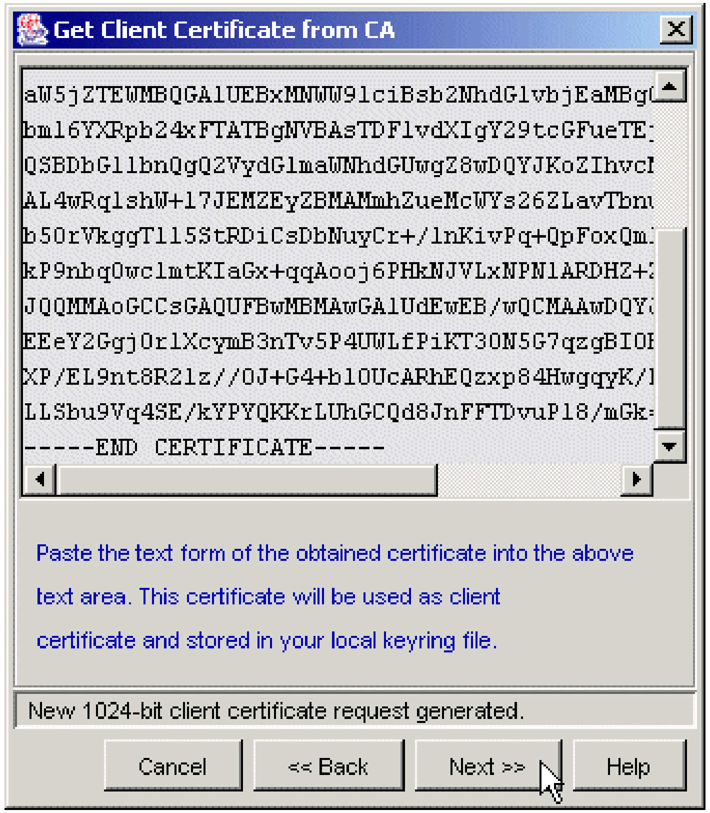
New 1024-bit client certificate request generated.

Copy and Paste the request into a CA's web site and let them sign the request.

You can paste the generated certificate into the text area on the following dialog box.

Setup a CA signed certificate - Wizard

10

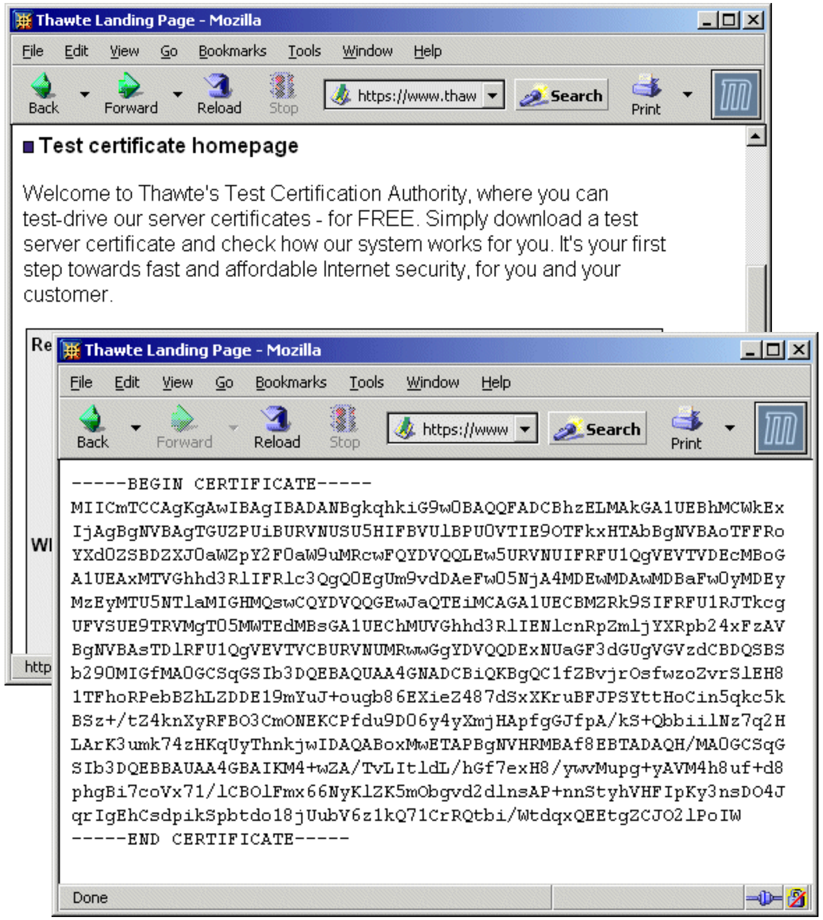


```
aW5jZTEWMBQGA1UEBxMNWW9lciBsb2NhdGlvbjEaMBg
bml6YXRpb24xFTATBgNVBAsTDFlvdXIgY29tcGFueTE
QSBDbGllbnQgQ2VydGlmaWNhdGUwgZ8wDQYJKoZIhvc
AL4wRqlshW+l7JEMZEyZBMAMmhZueMcWYs26ZLavTbn
b50rVkggTll5StRDiCsDbNuyCr+/lnKivPq+QpFoxQm
kP9nbq0wclmtKIaGx+qqAooj6PHkNJVLxNPNlARDHZ+
JQQMMAoGCCsGAQUFBwMBMAwGA1UdEwEB/wQCMAAwDQY
EEeY2Ggj0rlXcymB3nTv5P4UWLfPiKT30N5G7qzgBIO
XP/EL9nt8R2lz//0J+G4+bl0UcARhEQzxp84HwgqyK/
LLSbu9Vq4SE/kYPYQKKrLUhGCQd8JnFFTDvuPl8/mGk
-----END CERTIFICATE-----
```

Paste the text form of the obtained certificate into the above text area. This certificate will be used as client certificate and stored in your local keyring file.

New 1024-bit client certificate request generated.

11



Thawte Landing Page - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back
Forward
Reload
Stop
https://www.thaw
Search
Print
Re
Wl
http

Test certificate homepage

Welcome to Thawte's Test Certification Authority, where you can test-drive our server certificates - for FREE. Simply download a test server certificate and check how our system works for you. It's your first step towards fast and affordable Internet security, for you and your customer.

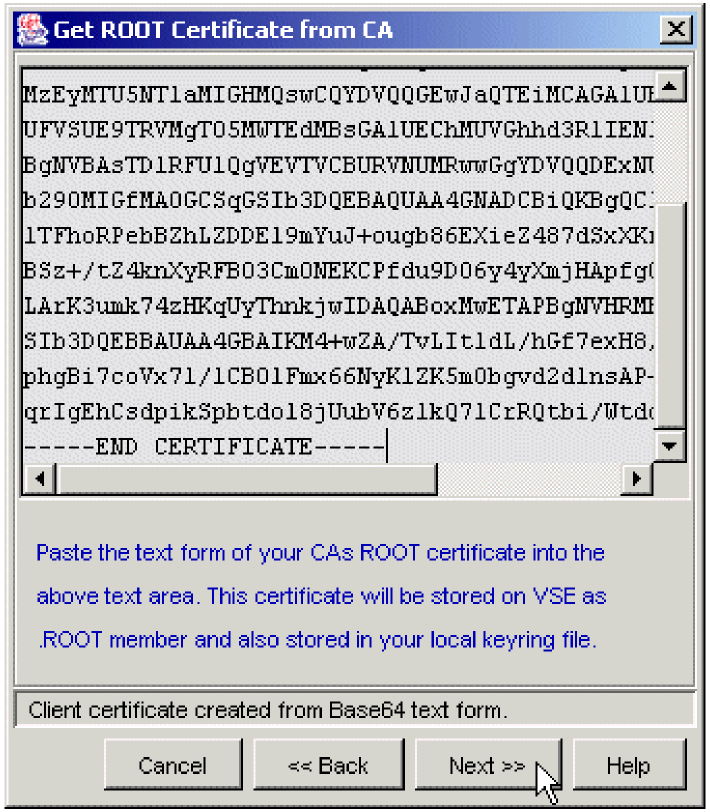
Thawte Landing Page - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back
Forward
Reload
Stop
https://www
Search
Print

```
-----BEGIN CERTIFICATE-----
MIICmTCCAgKgAwIBAgIBADANBgkqhkiG9w0BAQQFADCBhzELMAkGA1UEBhMCWkEx
IjAgBgNVBAgTGUZPUiBURVNUSU5HIFBVUlBPU0VTIE9OTFkxHTAbBgNVBAoTFFRo
YXd0ZSBDZXJ0aWZpY2F0aW9uMRcwFQYDVQQLEw5URVNUIFRFU1QgVEVTVDEcMBoG
A1UEAxMTVGhhd3RlIFRlc3QgQ0EgUm9vdDAeFw05NjA4MDEwMDAwMDBaFw0yMDEy
MzEyMTU5NTlaMIGHMQswCQYDVQQGEwJaQTEiMCAGA1UECBMZRk9SIFRFU1RJTkcg
UFVSUE9TRVMgT05MWTEdMBsGA1UEChMUVGhhd3RlIENlcnRpZmljYXRpb24xFzAV
BgNVBAsTDlRFU1QgVEVTVCBURVNUMRwwGgYDVQQDExNUaGF3dGUgVGVzdCBDQSBS
b290MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC1fZBvjrOsfwzoZvrSlEH8
1TFhoRPebBZhLZDDE19mYuJ+ougb86EXieZ487dSxXKruBFJPSYttHoCin5qkc5k
BSz+/tZ4knXyRFBO3CmONEKCPfdu9D06y4yXmjHApfgGJfpA/kS+QbbiilNz7q2H
LArK3umk74zHKqUyThnkjwIDAQABoxMwETAPBgNVHRMBAf8EBTADAQH/MA0GCSqG
SIb3DQEBBAUAA4GBAIKM4+wZA/TvLItldL/hGf7exH8/ywvMupg+yAVM4h8uf+d8
phgBi7coVx71/lCBOlFmx66NyKlZK5mObgvd2dlnsAP+nnStyhVHFIpKy3nsDO4J
qrIgEhCsdpikSpbtdo18jUubV6z1kQ71CrRQtbi/WtdqxQEEtgZCJO2lPoIW
-----END CERTIFICATE-----
```

Done

Setup a CA signed certificate - Wizard

12

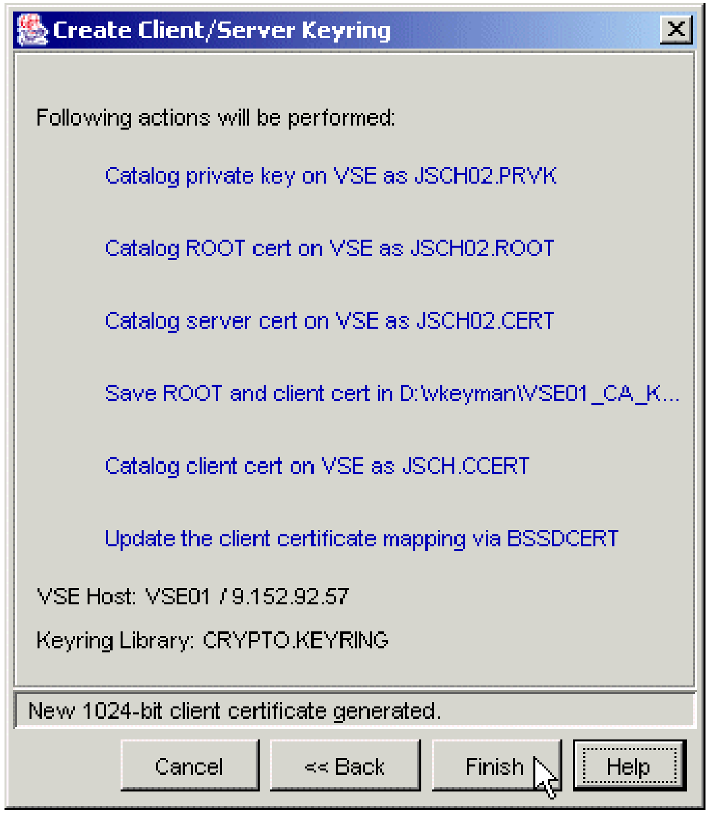


```
MzEyMTU5NTlaMIGHMQswCQYDVQQGEwJaQTEiMCAGA1UE
UFVSUE9TRVMgTO5MWTEdMBsGA1UEChMUVGhhd3RlIEN
BgNVBAsTD1RFU1QgVEVTVCBURVNUMRwwGgYDVQQDExN
b290MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC
lTFhoRPebBZhLZDDE19mYuJ+ougb86EXieZ487dSxXK
BSz+/tZ4knXyRFBO3CmONEKCPfdu9D06y4yXmjHApfg
LArK3umk74zHKqUyThnkjwIDAQABoxMwETAPBgNVHRM
SIb3DQEBBAUAA4GBAIKM4+wZA/TvLItldL/hGf7exH8,
phgBi7coVx7l/1CB01Fmx66NyK1ZK5m0bgvd2dlnsAP-
qrIgEhCsdpikSpbtdol8jUubV6zlkQ7lCrRQtbi/Wtd
-----END CERTIFICATE-----
```

Paste the text form of your CAs ROOT certificate into the above text area. This certificate will be stored on VSE as .ROOT member and also stored in your local keyring file.

Client certificate created from Base64 text form.

13



Following actions will be performed:

- Catalog private key on VSE as JSCH02.PRVK
- Catalog ROOT cert on VSE as JSCH02.ROOT
- Catalog server cert on VSE as JSCH02.CERT
- Save ROOT and client cert in D:\keyman\VSE01_CA_K...
- Catalog client cert on VSE as JSCH.CCERT
- Update the client certificate mapping via BSSDCERT

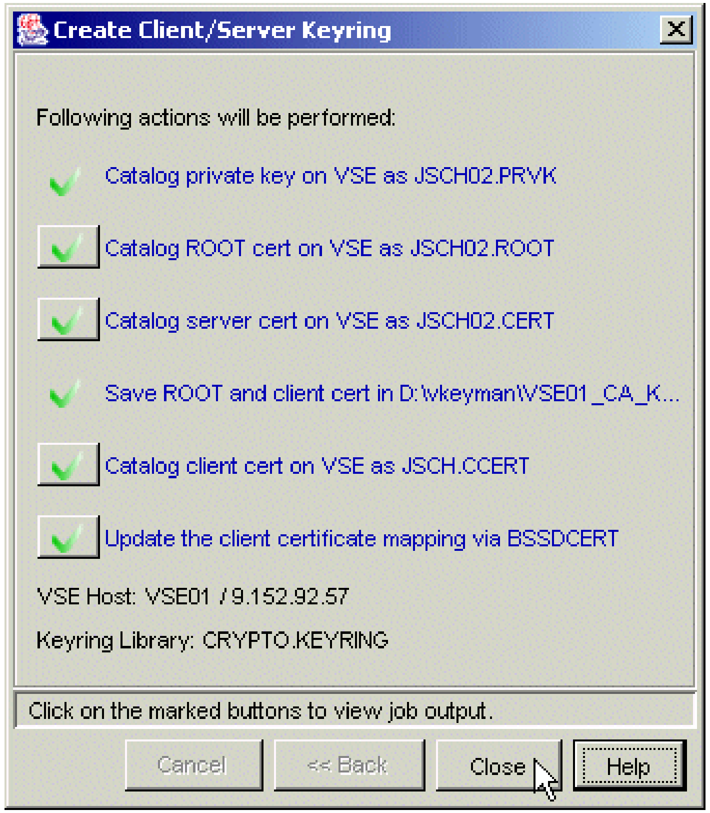
VSE Host: VSE01 / 9.152.92.57

Keyring Library: CRYPTO.KEYRING

New 1024-bit client certificate generated.

Setup a CA signed certificate - Wizard

14



Create Client/Server Keyring

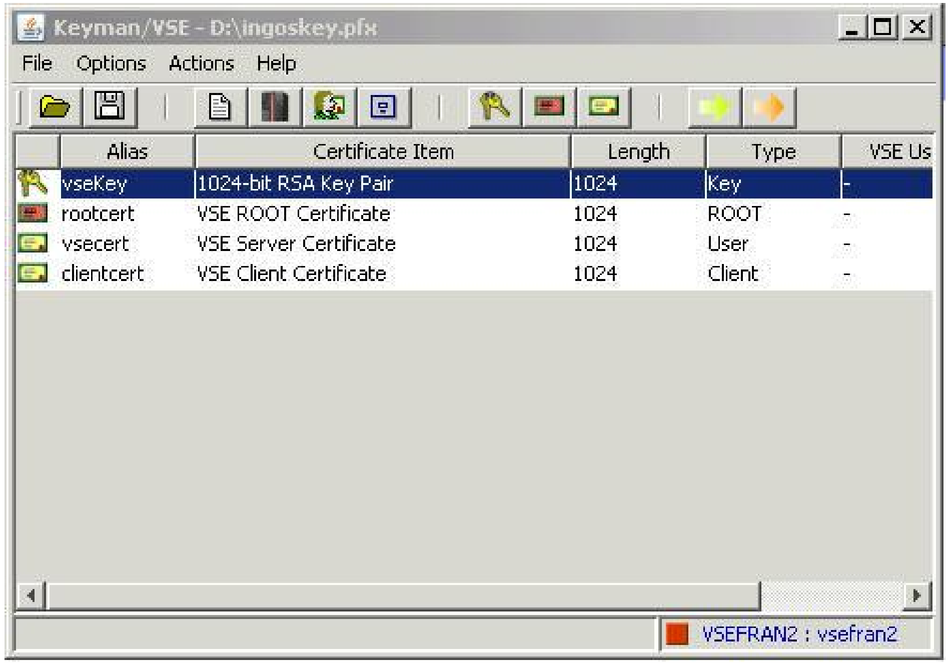
Following actions will be performed:

- ✓ Catalog private key on VSE as JSCH02.PRVK
- ✓ Catalog ROOT cert on VSE as JSCH02.ROOT
- ✓ Catalog server cert on VSE as JSCH02.CERT
- ✓ Save ROOT and client cert in D:\keyman\VSE01_CA_K...
- ✓ Catalog client cert on VSE as JSCH.CCERT
- ✓ Update the client certificate mapping via BSSDCERT

VSE Host: VSE01 / 9.152.92.57

Keyring Library: CRYPTO.KEYRING

Click on the marked buttons to view job output.



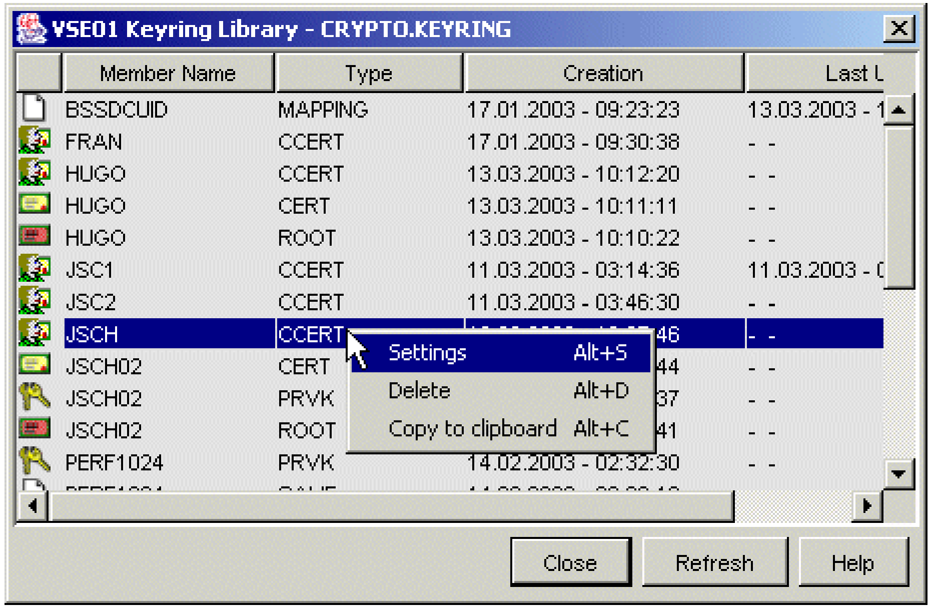
Keyman/VSE - D:\ingoskey.pfx

Alias	Certificate Item	Length	Type	VSE Us
vseKey	1024-bit RSA Key Pair	1024	Key	-
rootcert	VSE ROOT Certificate	1024	ROOT	-
vsecert	VSE Server Certificate	1024	User	-
clientcert	VSE Client Certificate	1024	Client	-

VSEFRAN2 : vsefran2

Where are keys and certificates stored on VSE ?

- Keys and certificates are stored on a VSE Library
  - Usually in CRYPTO.KEYRING
  - This library should be secured using the VSE security mechanisms (access protection)



VSE01 Keyring Library - CRYPTO.KEYRING

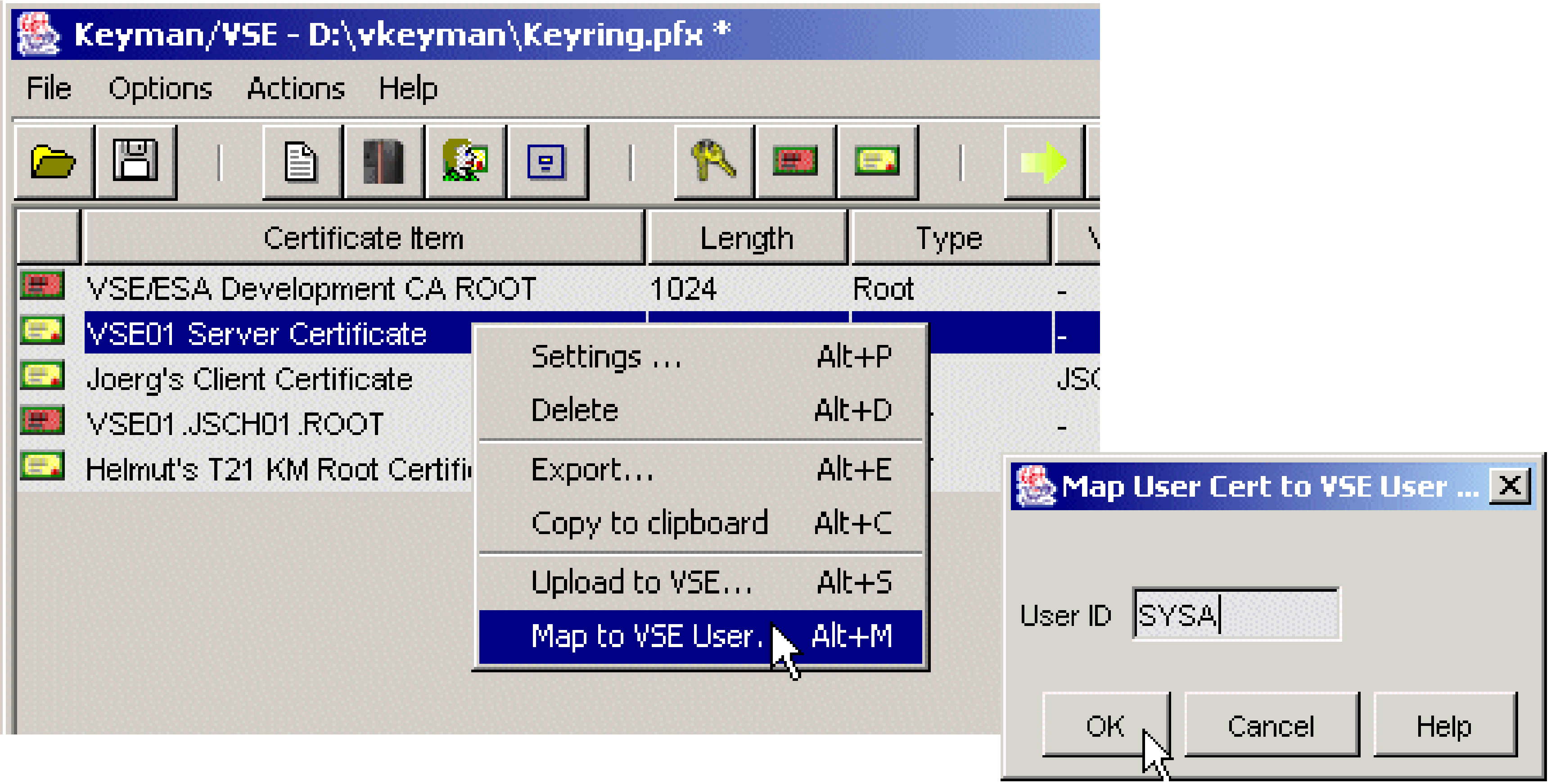
Member Name	Type	Creation	Last L
BSSDCUID	MAPPING	17.01.2003 - 09:23:23	13.03.2003 - 1
FRAN	CCERT	17.01.2003 - 09:30:38	- -
HUGO	CCERT	13.03.2003 - 10:12:20	- -
HUGO	CERT	13.03.2003 - 10:11:11	- -
HUGO	ROOT	13.03.2003 - 10:10:22	- -
JSC1	CCERT	11.03.2003 - 03:14:36	11.03.2003 - (
JSC2	CCERT	11.03.2003 - 03:46:30	- -
JSCH	CCERT	46	- -
JSCH02	CERT	44	- -
JSCH02	PRVK	37	- -
JSCH02	ROOT	41	- -
PERF1024	PRVK	14.02.2003 - 02:32:30	- -

- Member types:
  - .PRVK – Public/Private Key
  - .ROOT – Root Certificate
  - .CERT – Server Certificate
  - .CCERT – Client Certificate
  - BSSDCUID.MAPPING – Contains the User to Certificate mapping information

SSL with client authentication

- Server authentication means
  - The clients verifies the certificate received from the server
  - To make sure they are talking to the right server
- Client authentication means
  - The server verifies the certificates(s) received from the client(s)
  - To make sure only known clients can talk to the server
  - To do implicit logon by using the certificate (optional)
    - Map a user id to a certificate

Map a VSE user id to a client certificate



	Certificate Item	Length	Type	V
🟥	VSE/ESA Development CA ROOT	1024	Root	-
🟨	VSE01 Server Certificate			-
🟨	Joerg's Client Certificate			JSC
🟥	VSE01.JSCH01.ROOT			-
🟨	Helmut's T21 KM Root Certifi			

User ID: SYSA

SecureFTP

- The FTP protocol provides a easy and straight forward protocol for transferring files between systems on different platforms
  - Many installations rely on it to efficiently transmit critical files that can contain vital information such as customer names, credit card account numbers, social security numbers, corporate secrets and other sensitive information
  - FTP protocol transmits data without any authentication, privacy or integrity
- SecureFTP provides user authentication, privacy and integrity by using RSA digitally signed certificates, DES encryption and SHA-1 secure hash functions
  - SecureFTP is integrated into TCP/IP for VSE with z/VSE V4.1 (at no additional charge) or offered as separately priced product by CSI
- How to setup Secure FTP with VSE:
  ftp://ftp.software.ibm.com/eserver/zseries/zos/vse/pdf3/How_to_setup_SecureFTP_with_VSE.pdf

Hardware Crypto Support on System z and VSE

by release

	z/VSE 4.2	z/VSE 4.1	z/VSE 3.1	VSE/ESA 2.7	VSE/ESA 2.6
PCICA	Yes	Yes	Yes	Yes	-
CEX2C	Yes	Yes	Yes	-	-
CPACF	Yes	Yes	Yes	-	-
CEX2A	Yes	Yes	Yes	-	-
PCIXCC	Yes	Yes	-	-	-

	prior z800	z800	z900	z890	z990	Z9	z10
PCICA	-	Yes	Yes	Yes	Yes	-	-
PCIXCC	-	-	-	Yes	Yes	-	-
CEX2C	-	-	-	Yes	Yes	Yes	Yes
CPACF	-	-	-	Yes	Yes	Yes	Yes
CEX2A	-	-	-	-	-	Yes	Yes

by server

CEX2C = Crypto Express2 in coprocessor mode
CEX2A = Crypto Express2 in accelerator mode
See: http://www.ibm.com/systems/z/security/cryptography.html

VSE Hardware Configuration

- VSE hardware configuration not necessary for crypto hardware
  - No IOCDS definition in VSE
  - No device type
  - No ADD statement
  - You may have to define the devices in the HMC (LPAR) or z/VM directory
- Use of crypto hardware is transparent to end users and even TCP/IP applications
  - But use of crypto hardware can be disabled via TCP/IP SOCKOPT phase
- How to setup cryptographic hardware for VSE:
  ftp://ftp.software.ibm.com/eserver/zseries/zos/vse/pdf3/How_to_setup_crypto_hardware_for_VSE.pdf

HW-Crypto related console messages

§ System with crypto hardware

```
FB 0095 1J023I FOUND A CRYPTO EXPRESS2 CARD AT DEVICE INDEX 0
FB 0095 1J023I FOUND A CRYPTO EXPRESS2 CARD AT DEVICE INDEX 1
FB 0095 1J014I FOUND A PCICA CARD AT DEVICE INDEX 6
FB 0095 1J014I FOUND A PCICA CARD AT DEVICE INDEX 7
FB 0095 1J005I HARDWARE CRYPTO ENVIRONMENT INITIALIZED SUCCESSFULLY.
FB 0095 1J006I USING CRYPTO DOMAIN 0
FB 0095 1J022I CPU CRYPTOGRAPHIC ASSIST FEATURE AVAILABLE.
```

§ System without crypto hardware

```
FB 0093 1J020W THERE WAS NO PCICA OR CRYPTO EXPRESS2 CARD
FB 0093        FOUND. HARDWARE CRYPTO NOT AVAILABLE.
```

HW-Crypto status display

```
msg fb,data=status=cr
AR 0015 1I40I  READY
FB 0011 BST223I CURRENT STATUS OF THE SECURITY TRANSACTION SERVER:
FB 0011 ADJUNCT PROCESSOR CRYPTO SUBTASK STATUS:
FB 0011   AP CRYPTO SUBTASK STARTED .......... : YES
FB 0011   MAX REQUEST QUEUE SIZE ............. : 1
FB 0011   MAX PENDING QUEUE SIZE ............. : 1
FB 0011   TOTAL NO. OF AP REQUESTS ........... : 1234
FB 0011   NO. OF POSTED CALLERS .............. : 1234
FB 0011   AP CRYPTO POLLING TIME (1/300 SEC).. : 1
FB 0011   AP CRYPTO TRACE LEVEL .............. : 3
FB 0011   ASSIGNED APS : PCICC / PCICA ....... : 0 / 0
FB 0011                  CEX2C / CEX2A ....... : 1 / 2
FB 0011                  PCIXCC .............. : 0
FB 0011     AP   0 : CEX2C   - ONLINE
FB 0011     AP   4 : CEX2A   - ONLINE
FB 0011     AP   9 : CEX2A   - ONLINE
FB 0011   ASSIGNED AP QUEUE (CRYPTO DOMAIN)... : 6
FB 0011 CPU CRYPTOGRAPHIC ASSIST FEATURE:
FB 0011   CPACF AVAILABLE .................... : YES
FB 0011   INSTALLED CPACF FUNCTIONS:
FB 0011     DES, TDES-128, TDES-192, SHA-1
FB 0011     AES-128
FB 0011     PRNG, SHA-256
FB 0011 END OF CPACF STATUS
```

Crypto HW exploitation in VSE

- Crypto cards are only used for RSA acceleration
  - RSA decrypt/encrypt for SSL session initiation
  - RSA encrypt for signing of certificates (CIALCREQ)
- CPACF
  - Acceleration of symmetric algorithms:
    DES, TDES, AES-128 (z9 only), SHA-1
  - Used at
    - SSL data transfer
    - CIAL functions in TCP/IP
- Usage is transparent for TCP/IP applications
  - If Crypto HW is available, it will be used. If not available, the SW implementation (as part of TCP/IP) will be used
  - You can disable the use of Crypto HW via a setting in $SOCKOPT Phase

Crypto HW exploitation in VSE

- HW Crypto Functions that are not exploited in VSE
  - Special functions available in Coprocessor-Modus
    - **RSA Key-Generation**
      - RSA keys could be generated directly on VSE, no workstation tool would be required
    - **Secure Key functions**
      - PIN functions
      - Symmetric Key Import / Export (Key Transport)
    - **Special functions for banking-software**
      - ANSI X9.17 Standard: Key generate, export, import

- Requirements are welcome !

Secure Key vs. Clear Key

- Different way of managing, storing and usage of keys
  - Keys reside unencrypted (clear) in the file system (“Clear Key”)
  - Keys reside encrypted (TDES with fixed key) in the file system
    - è That’s how VSE works today
  - Keys reside encrypted (using a “Secure Master Key“) in the file system
    - The Master Key is stored in the hardware
    - Secure master key entry via TKE or Dialogs
    - Crypto operations are done in main storage, i.e. data keys are visible (unencrypted) in main storage for a very short time
    - Crypto operations are done on a coprocessor card, i.e. data keys will never reside unencrypted in the main storage
      - è Required for banking applications, e.g. PIN Verification
      - è Supported by z/OS ICSF

CryptoVSE API

- Native cryptographic API
  - Can also be used directly from within COBOL programs
- Provides cryptographic services:
  - Data encryption
    - DES
    - Triple DES
    - AES
    - RSA PKCS #1
  - Message Digest
    - MD5
    - SHA-1
  - Digital Signatures
    - RSA PKCS #1 with SHA1 or MD5
  - Message Authentication
    - HMAC
- Uses Hardware Crypto functions transparently when available

Customer Data Protection Requirements

- Regulatory requirements driving need for greater data security, integrity, retention/auditability, and privacy
- Severe business impacts caused by loss or theft of data including financial liability, reputation damage, legal/compliance risk
- Increasing need to share data securely with business partners and maintain backups at remote locations
- Need to reduce complexity and improve processes around enterprise encryption management
- Need ability to cost effectively encrypt large quantities of tape data

Data Center

In Transit

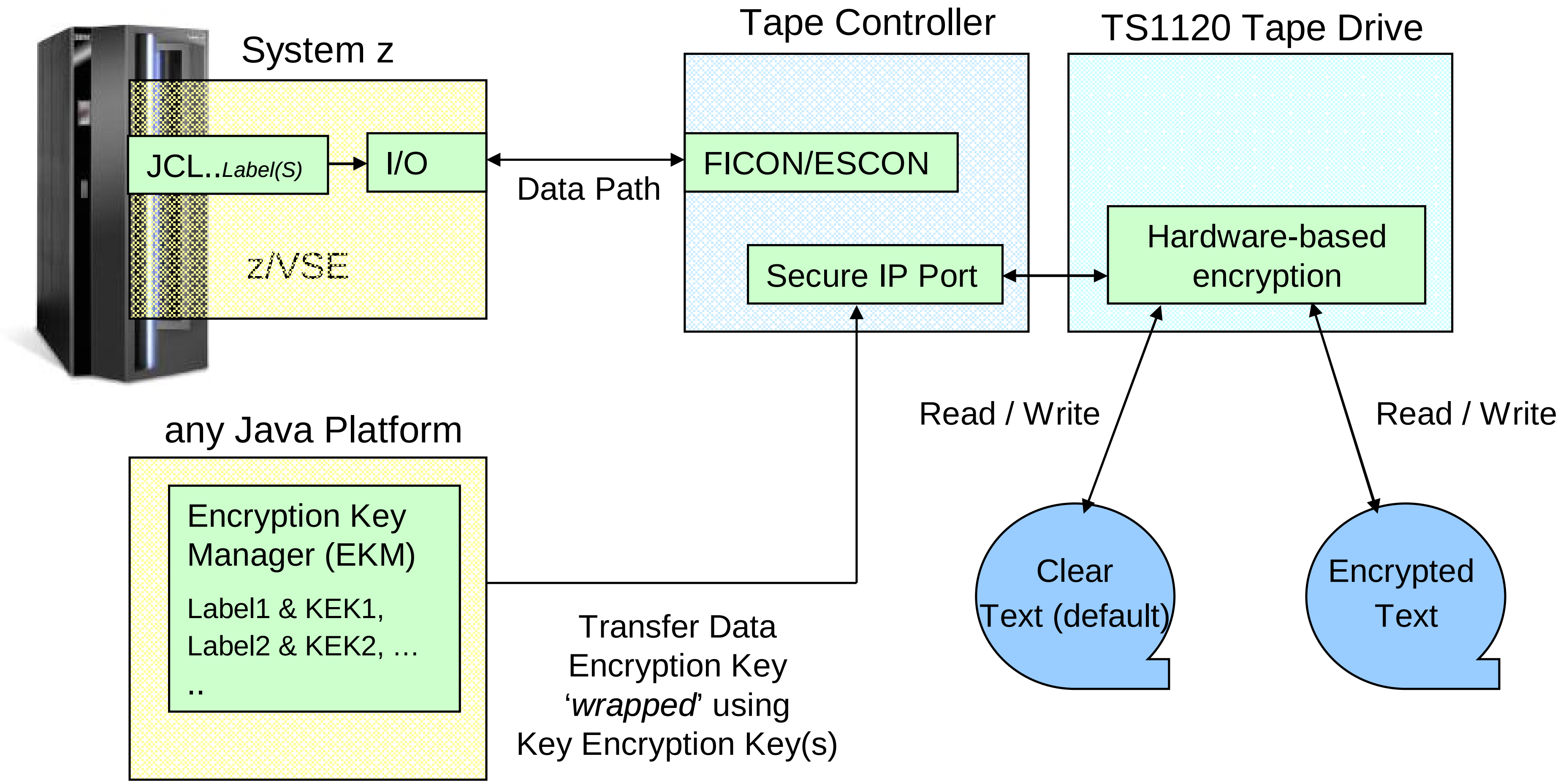
Secondary Site

Business Partners

IBM Tape Encryption – TS1120

- The IBM System Storage TS1120 Tape Drive has been enhanced to provide drive based data encryption
- A new, separate IBM Encryption Key Manager component for the Java Platform (Encryption Key Manager) program is also being introduced:
  - supports the generation and communication of encryption keys for the tape drives across the enterprise.
- *New: Support is now available for z/VSE V4 and V3:*
  - *z/VSE V4.1: DY46682 (UD53141 and UD53142)*
  - *z/VSE V3.1: DY46685 (UD53143,UD53144, UD53146) and PK43473 (UK24398)*
  - *z/VM: VM64062 (UM32012)*
  - *DITTO: PK44172 - With this Apar, DITTO/ESA for VSE supports tape encryption interactively and via standard VSE JCL in BATCH mode*

IBM Tape Encryption – TS1120



IBM Tape Encryption – TS1120

```
// JOB ENCRYPT
// ASSGN SYS005,480,03
// KEKL UNIT=480,KEKL1='MYKEKL1',KEM1=L,KEKL2='MYKEKL2',KEM2=L
// EXEC LIBR
  BACKUP LIB=PRD2 TAPE=SYS005
/*
/&
```

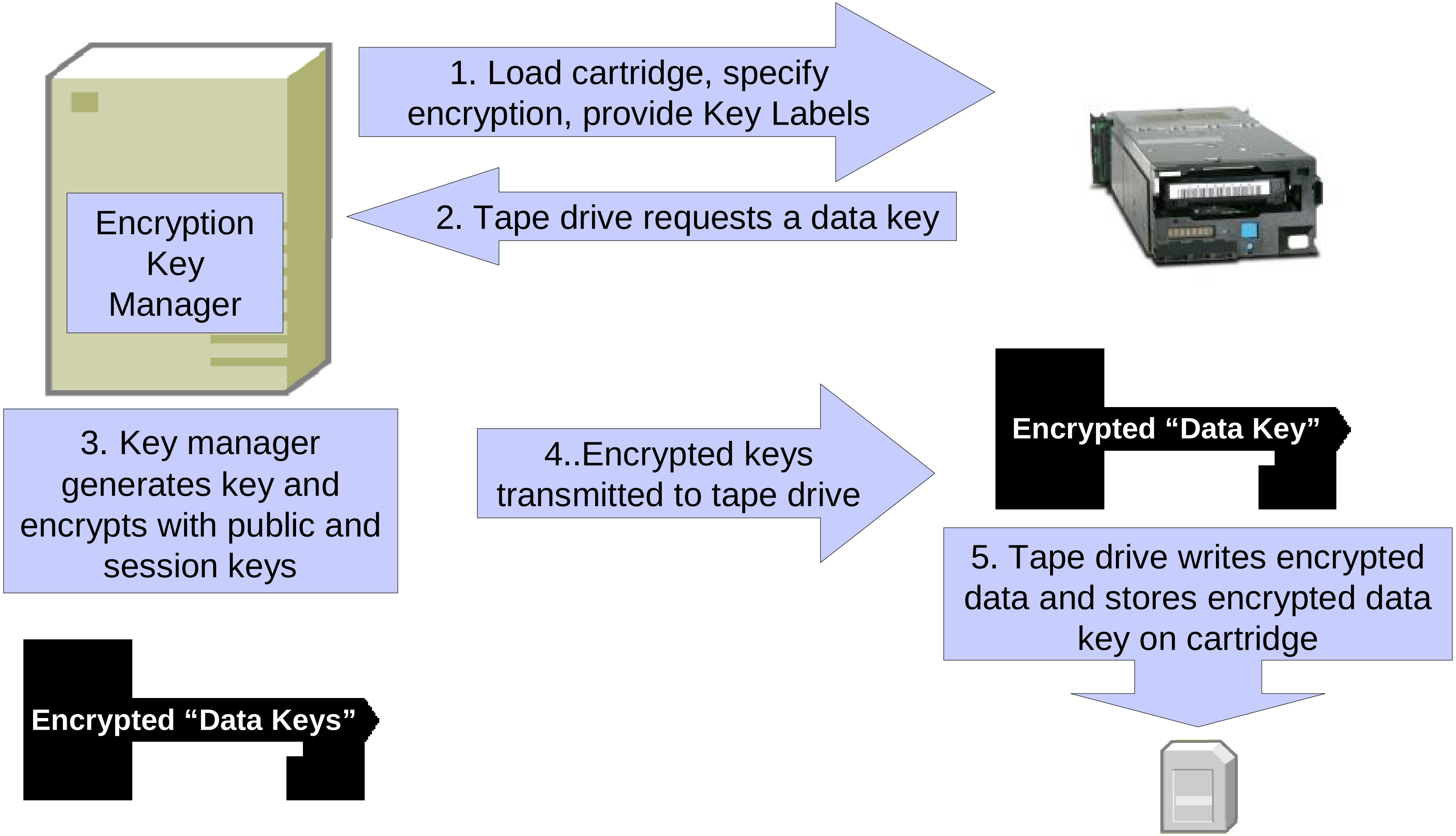
encryption mode
(03=write)

key label1
(name of the 1. KEK-key in EKM)

encoding mechanism
(L=Label, H=Hash)

- The Data-Key can be encrypted using 2 different public keys (KEK = Key Encrypting Keys), to be able to send the tape to 2 different receivers
- More info can be found in the *z/VSE 4.1 Administration* manual (VSE Homepage)

IBM Tape Encryption – TS1120



Encryption Key Manager

1. Load cartridge, specify encryption, provide Key Labels

2. Tape drive requests a data key

3. Key manager generates key and encrypts with public and session keys

Encrypted "Data Keys"

4..Encrypted keys transmitted to tape drive

Encrypted "Data Key"

5. Tape drive writes encrypted data and stores encrypted data key on cartridge

IBM Tape Encryption – TS1120

- Considerations and Restrictions:
  - A tape can either contain encrypted data or unencrypted data
  - If you encrypt the first file on the tape, all subsequent files will also be encrypted using the same key
    - Important for multi file tapes
  - If you send an encrypted tape to a business partner, the other side will also require a TS1120 to be able to read the tape

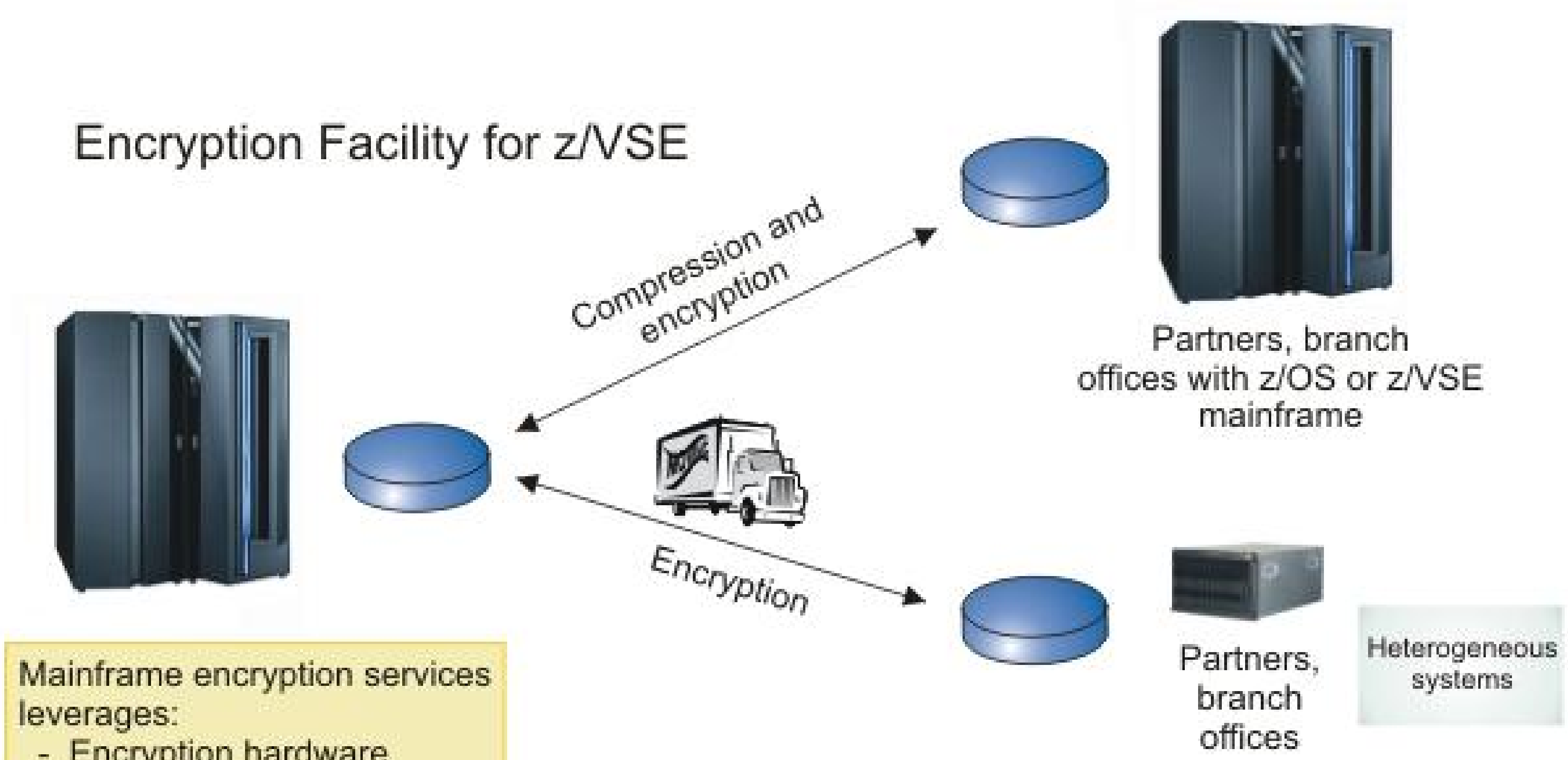
IBM Tape Encryption – TS1120 - Summary

- Hardware-based encryption
  - No host cycles used
- Designed for high volume backup
- Encryption Key Manager (EKM) on a Java platform
  - for centralized key management
  - with SSL connection between tape controller and EKM
- Encryption option specified in VSE via JCL commands
  - // ASSGN …
  - // KEKL …

New: Encryption Facility for z/VSE

- IBM Encryption Facility for z/VSE V1.1 can help you:
  - Secure business and customer data
  - Address regulatory requirements
  - Protect data from loss and inadvertent or deliberate compromise
  - Enable sharing of sensitive information across platforms with partners, vendors, and customers
  - Enable decrypting and encrypting of data to be exchanged between z/VSE and non-z/VSE platforms
- The Encryption Facility for z/VSE V1.1 is packaged as an optional, priced feature of VSE Central Functions V8.1 (5686-CF8-40).
  - Documentation in z/VSE 4.1.1 Administration book, Chapter 43
  - Available since November 30, 2007

New: Encryption Facility for z/VSE


Encryption Facility for z/VSE
Compression and encryption
Encryption
Partners, branch offices with z/OS or z/VSE mainframe
Partners, branch offices
Heterogeneous systems

Mainframe encryption services leverages:
- Encryption hardware
- Client / Server based key management

New: Encryption Facility for z/VSE

- The Encryption Facility for z/VSE V1.1 uses the same data format as the Encryption Services feature in Encryption Facility for z/OS V1.1 and V1.2 (5655-P97)
  - Called ‚Encryption Facility System z format‘
- It allows you to exchange encrypted files between
  - your internal mainframe data centers
  - you and your external business partners and vendors

- To decrypt an encrypted file, you must have installed any of the following:
  - Encryption Facility for z/VSE feature
  - Encryption Facility for z/OS Encryption Services feature (using System z format)
  - The no-charge Encryption Facility for z/OS Client Web download
    - either Java-based client
    - or Decryption Client for z/OS

New: Encryption Facility for z/VSE

Possible choices:

	Decrypt data using System z format with:			
Encrypt data using System z format with:	Encryption Services feature of EF for z/OS	Encryption Facility for z/VSE	Decryption Client for z/OS	Java-based Client
Encryption Services feature of EF for z/OS	**Yes**	**Yes**	**Yes**	**Yes**
EnecryptinFacility for z/VSE	**Yes**	**Yes**	**Yes**	**Yes**
Java-based Client	**Yes**	**Yes**	**No (*)**	**No (*)**

Note: The terms and conditions for the no-charge Encryption Facility for z/OS Client only allow the use of the Encryption Facility for z/OS Client for decrypting information or data that was encrypted by IBM's Encryption Facility for z/OS or IBM's Encryption Facility for z/VSE, or for encrypting information or data to be decrypted by IBM's Encryption Facility for z/OS or IBM's Encryption Facility for z/VSE.

New: Encryption Facility for z/VSE

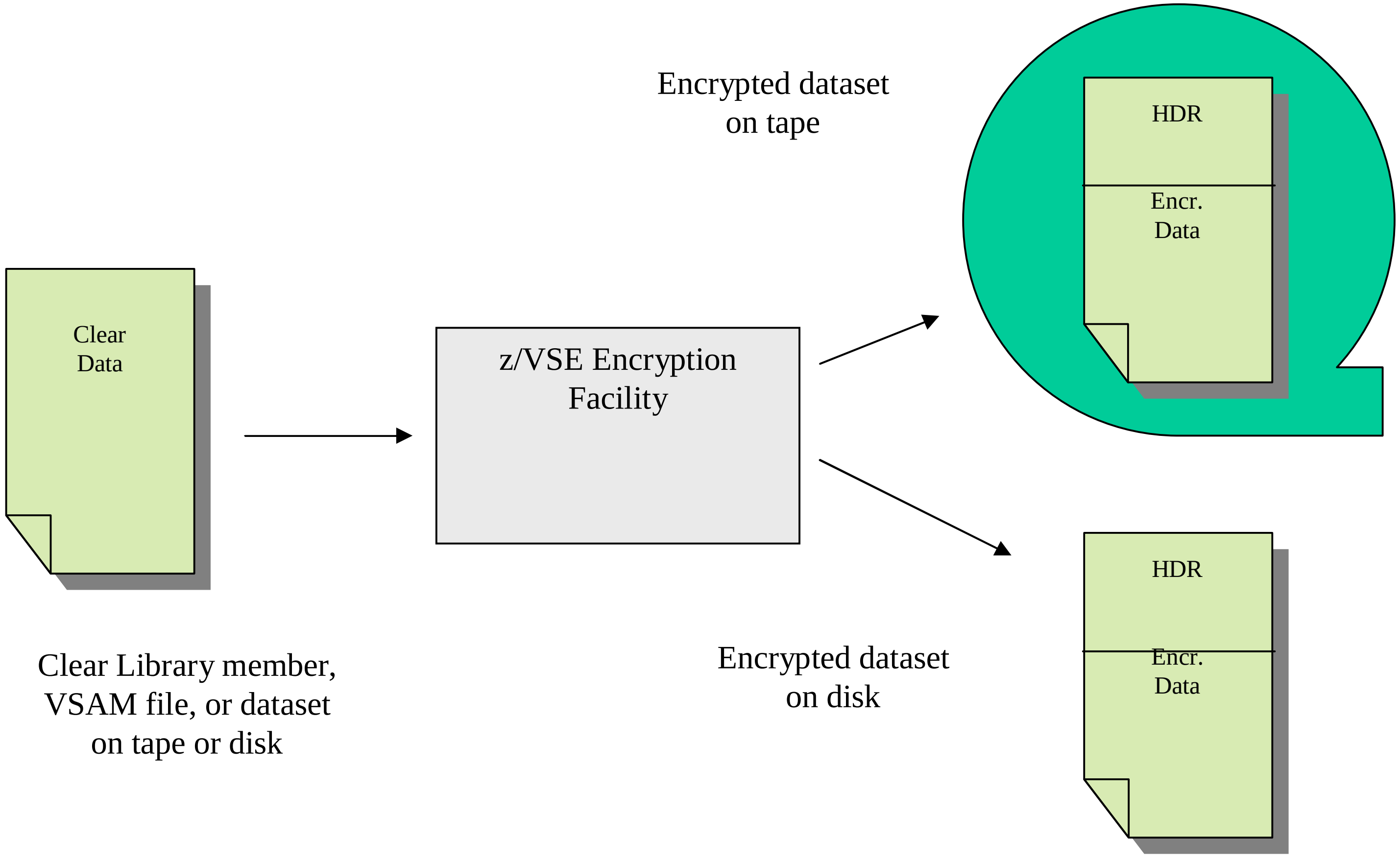
- Encryption Facility for z/VSE supports
  - Password-based encryption of session keys
  - Data encryption with a randomly generated symmetric session key using AES-128 or Triple-DES algorithms
  - Asymmetric encryption of randomly generated symmetric keys using the RSA algorithm with key lengths of 512, 1024 and 2048-bit
  - Encryption of single SAM files, VSAM files, or VSE Library members
  - Encryption of virtual or real tapes
- Support of hardware-accelerated compression before encryption
- Encryption of complete backups made with any backup tool either from IBM or vendors
- Output of encrypted data on disk, virtual tape, or real tape
  - As sequential file (SAM or VSAM)

New: Encryption Facility for z/VSE

- Hardware Requirements:
  - For the PASSWORD option, use CPACF only.
  - For the Clear-TDES and Clear-AES-128 (no ENCTDES), use CPACF only.
  - For RSA keys (bit length 2048), use one of the following:
    - Crypto Express2-accelerator mode (CEX2A)
    - Crypto Express2-coprocessor mode (CEX2C)
    - PCIX Cryptographic Coprocessor (PCIXCC)

- Software requirements
  - z/VSE 4.1 with APAR DY46717 (PTF UD53196)
  - For public encryption, TCP/IP for VSE/ESA V1.5E, or higher, is required.
  - For RSA keys (bit length 1024) TCP/IP for VSE/ESA V1.5E, or higher, is required.
  - For RSA keys (bit length 2048) refer to the Hardware requirements.

New: Encryption Facility for z/VSE - Encryption of a single file



Clear Data

z/VSE Encryption Facility

Encrypted dataset on tape

HDR

Encr. Data

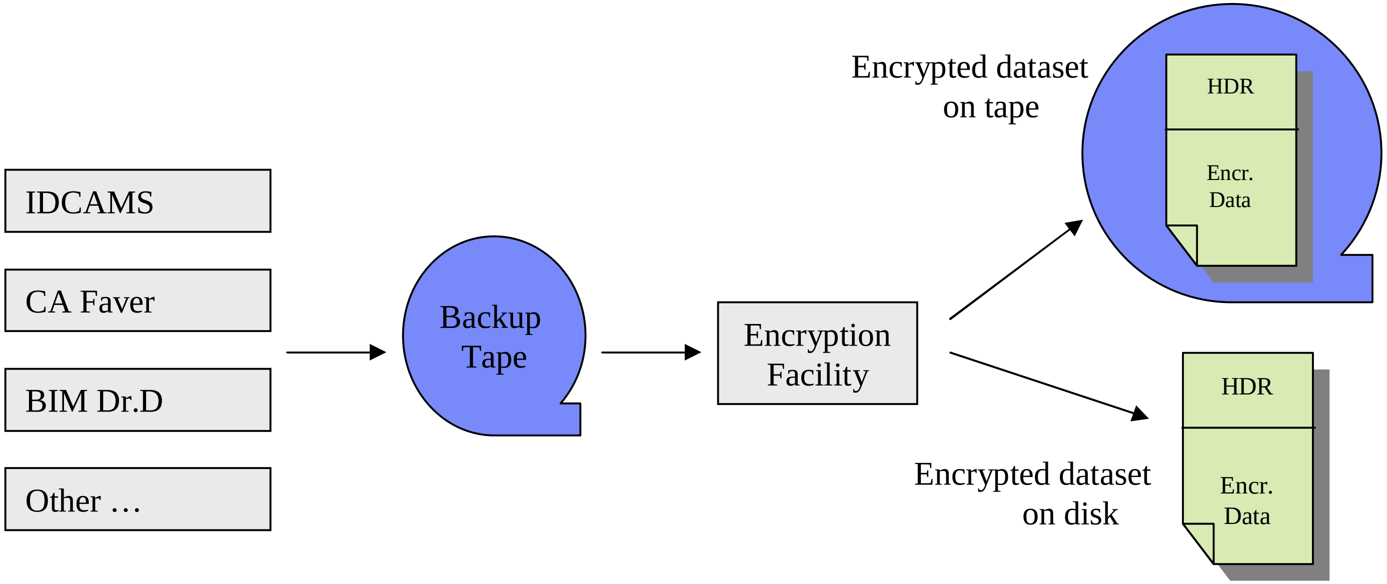
Encrypted dataset on disk

HDR

Encr. Data

Clear Library member, VSAM file, or dataset on tape or disk

New: Encryption Facility for z/VSE - Encryption of a complete backup



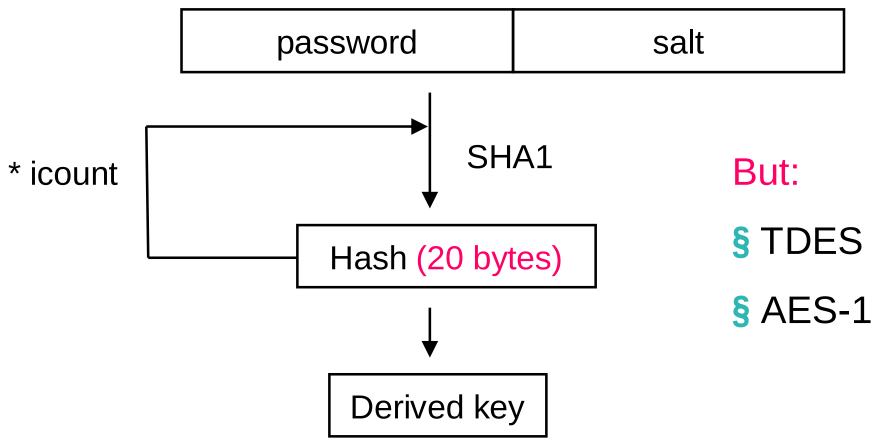
- Any proprietary backup tape can be encrypted and written to a second tape or to disk.
- Note that the complete input tape results in just one encrypted dataset, which resides on tape or disk.

New: Encryption Facility for z/VSE - Password-based encryption (PBE)

- Encryption key (data key) is generated from
  - the given secret password (8 … 32 characters)
  - iteration count, and
  - a 8-byte random number (the “salt”), which is different for each encryption process.
- The iteration count and salt value are stored in the encrypted dataset header.
  - icount and salt are not secret
  - When encrypting the same data twice with the same password and iteration count, the resulting encrypted data will be completely different, because of the randomly created salt value.
- No need to deal with keys, but
- Need to manage/archive passwords
  - Many free tools available, e.g.
  - KeePass : http://keepass.sourceforge.net/

New: Encryption Facility for z/VSE - PBE: Example for generating a key

- Example of a “Password-based key derivation function”
- PBKDF1(password, salt, iteration_count, dkLen)
- Disadvantage:
  - Derived key length (dkLen) limited to output of underlying hash function (MD5 = 16 bytes, SHA-1 = 20 bytes)
  - Used today only for compatibility with older applications
- Described in RFC 2898
- Process:



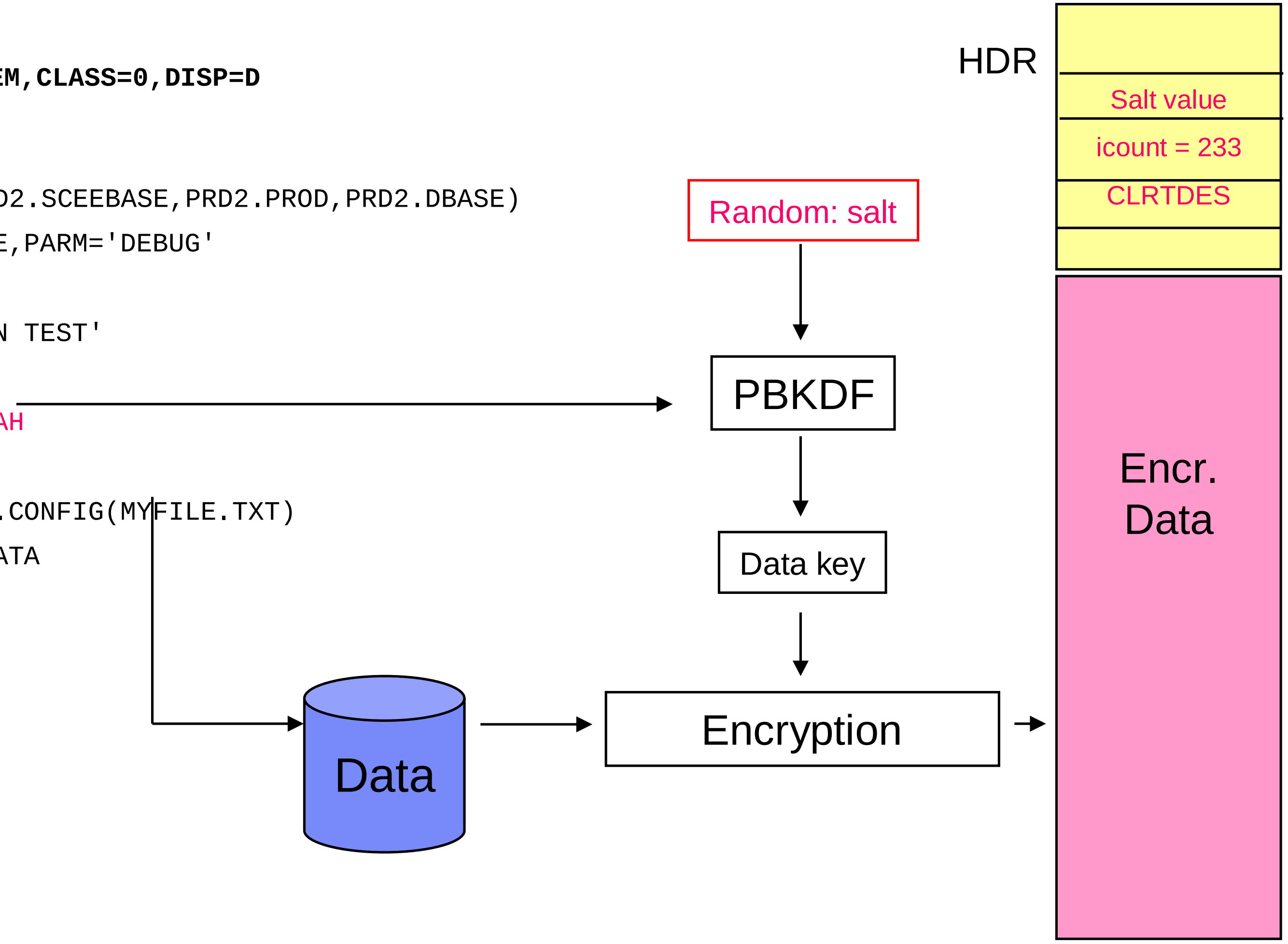
Note: this is not exactly the process used in Encryption Facility for z/VSE. It’s only an example.

But:

- TDES key = 3 * 8 bytes = 24 + 8 bytes ICV = 32
- AES-128 key = 16 bytes + 16 bytes ICV = 32

New: Encryption Facility for z/VSE - PBE: Job example for encryption

```
* $$ JOB JNM=ENCMEM,CLASS=0,DISP=D
  // JOB ENCMEM
  // LIBDEF
      *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
  // EXEC IJBEFVSE,PARM='DEBUG'
  ENCRYPT
  DESC='ENCRYPTION TEST'
  CLRTDES
  PASSWORD=BLAHBLAH
  ICOUNT=233
  CLRFILE=DD:PRD2.CONFIG(MYFILE.TXT)
  ENCFILE=DD:ENCDATA
  /*
  /&
  * $$ EOJ
```

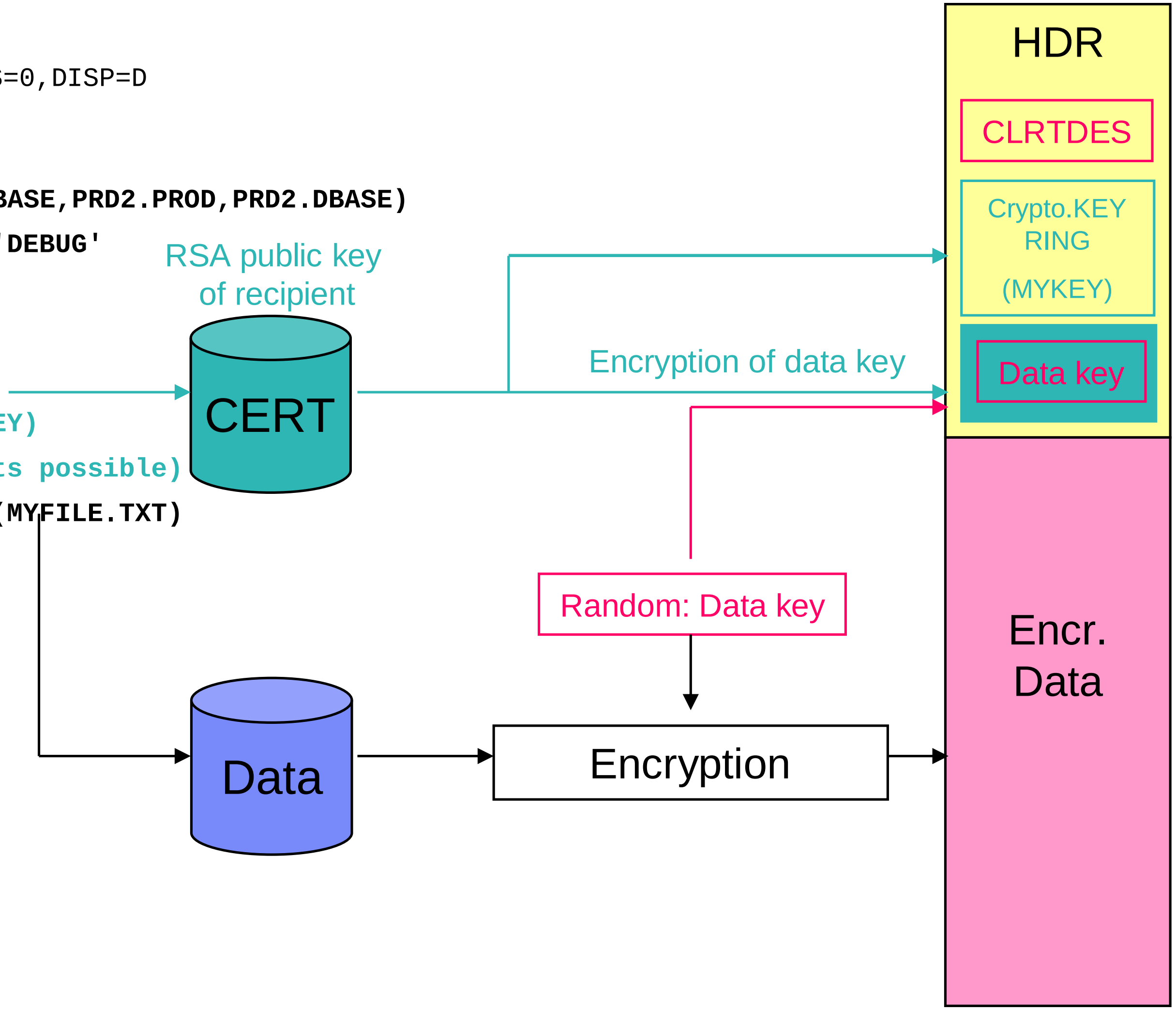


New: Encryption Facility for z/VSE - Public-key encryption (PKE)

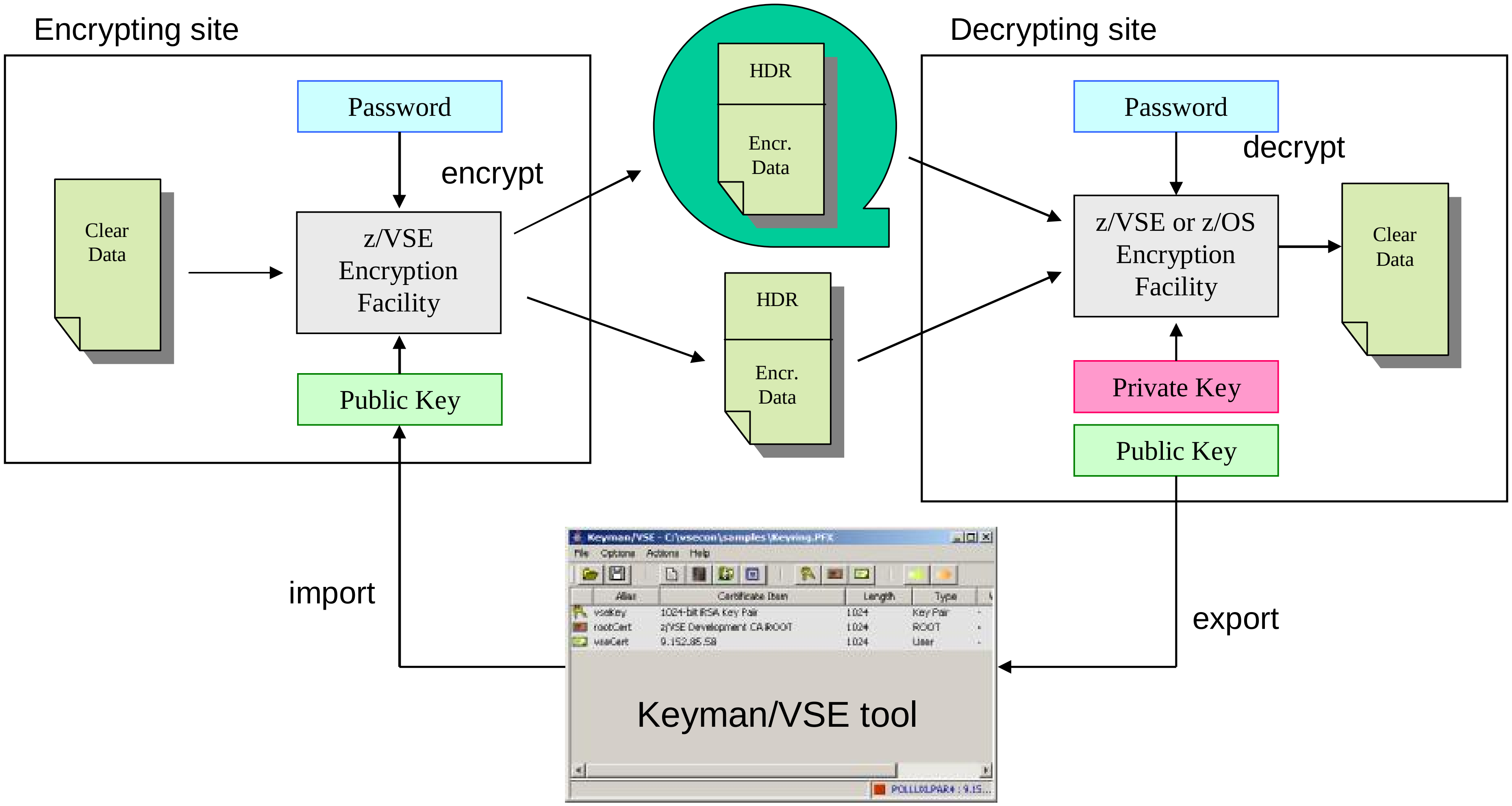
- Encryption key (data key) is randomly generated
- Data key is then encrypted with the public key of the recipient of the encrypted data
  - Needs a Crypto Express2 or PCIXCC card for 2048 bit keys
  - Crypto cards are transparently used also for 1024 bit keys when available
- Data key is put into the encrypted dataset together with the encrypted data
- Only one recipient is able to decrypt the data key and thus, the encrypted data, using the corresponding private key
- Need to manage / exchange public RSA keys
  - Can be done with the Keyman/VSE tool

New: Encryption Facility for z/VSE - PKE: Job example for encryption

```
* $$ JOB JNM=ENCMEM,CLASS=0,DISP=D
  // JOB ENCMEM
  // LIBDEF
      *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
  // EXEC IJBEFVSE,PARM='DEBUG'
  ENCRYPT
  DESC='ENCRYPTION TEST'
  CLRTDES
  RSA=CRYPTO.KEYRING(MYKEY)
  (up to 16 RSA statements possible)
  CLRFILE=DD:PRD2.CONFIG(MYFILE.TXT)
  ENCFILE=DD:ENCDATA
  /*
  /&
  * $$ EOJ
```



New: Encryption Facility for z/VSE - PBE and PKE scenario



Keyman/VSE - C:\vsecon\samples\Keyring.PFX

	Alias	Certificate Item	Length	Type
	vseKey	1024-bit RSA Key Pair	1024	Key Pair
	rootCert	z/VSE Development CA ROOT	1024	ROOT
	vseCert	9.152.85.58	1024	User

POLLUXPAR4 : 9.15...

New: Encryption Facility for z/VSE - Customer value

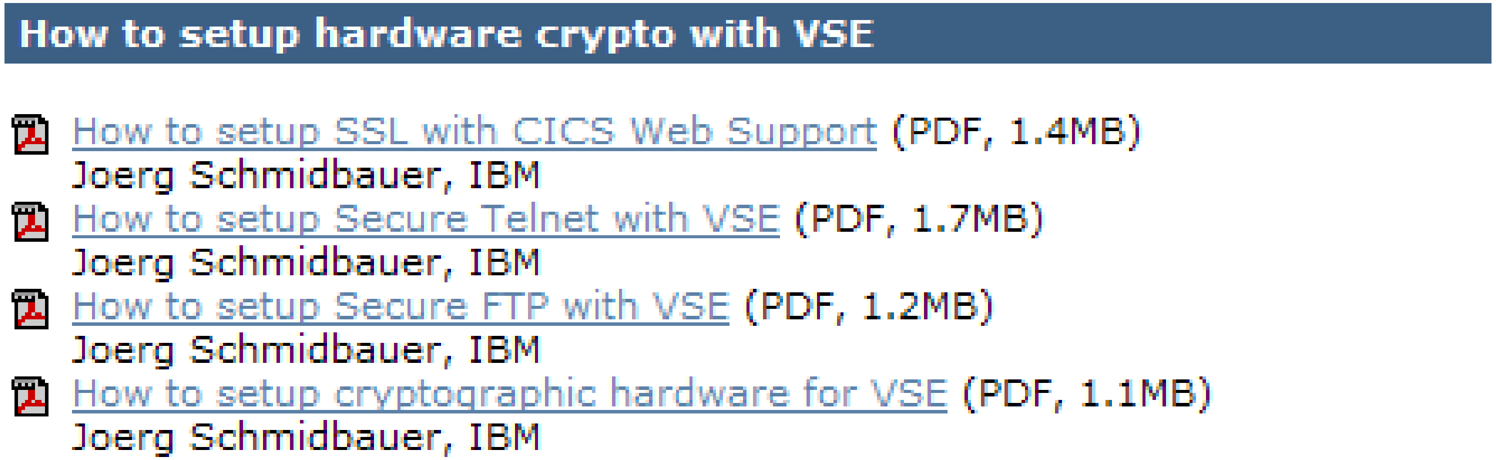
- **No special tape hardware requirements (e.g. TS1120)**
  - But exploits IBM crypto hardware (crypto cards and CPACF)
- **Host-based utility, no additional client/server workstations**
- **Easy to use**
  - No special setup necessary for password-based encryption
- **Supports all VSE data formats: single files and complete tape backups (LIBR, IDCAMS, POWER, etc.)**
- **Supports even proprietary vendor backup formats**
- **Encrypted datasets and tapes can easily be exchanged between business partners even on non z platforms**
  - Password-based
  - Public-key based

Other ways to encrypt your backups or tapes

- Can be done using VTAPE
  - Create a backup on a remote virtual tape
  - Store the tape image on an encrypted medium
    - Encrypted file system or directory (e.g. EcryptFS on Linux)
    - Use encryption tools (e.g. TrueCrypt)
    - Use Tivoli Storage Manager to store the backup data
- Encrypt data in applications
  - Use CryptoVSE API to encrypt the data
    - Uses Hardware Crypto Support if available

New technical articles on VSE homepage

http://www.ibm.com/servers/eserver/zseries/zvse/documentation/security.html#howto



How to setup hardware crypto with VSE

- How to setup SSL with CICS Web Support (PDF, 1.4MB)
  Joerg Schmidbauer, IBM
- How to setup Secure Telnet with VSE (PDF, 1.7MB)
  Joerg Schmidbauer, IBM
- How to setup Secure FTP with VSE (PDF, 1.2MB)
  Joerg Schmidbauer, IBM
- How to setup cryptographic hardware for VSE (PDF, 1.1MB)
  Joerg Schmidbauer, IBM

Related Documentation

- VSE Homepage
  http://www.ibm.com/servers/eserver/zseries/zvse/
- Keyman/VSE tool and VSE Connector Client
  http://www.ibm.com/servers/eserver/zseries/zvse/downloads/
- Encryption Facility for z/OS
  http://www.ibm.com/servers/eserver/zseries/zos/encryption_facility/
- IBM Encryption Facility for z/OS Java Client
  http://www.ibm.com/servers/eserver/zseries/zos/downloads/#efclient
- IBM PCI Cryptographic Accelerator (PCICA)
  http://www.ibm.com/security/cryptocards/pcica.shtml
- IBM Crypto Express2 (CEX2)
  http://www.ibm.com/systems/z/security/cryptography.html
- CP Assist for Cryptographic Function (CPACF)
  http://www.ibm.com/systems/z/security/cryptography.html
- IBM Security Products – Overview
  http://www.ibm.com/security/products/
- KeePass Password Safe – a free Open Source Password Manager for many operating systems
  http://keepass.sourceforge.net/

Questions ?

?