S11 – Encryption Facility for z/VSE, Verschlüsselung im und rund ums z/VSE

Jörg Schmidbauer

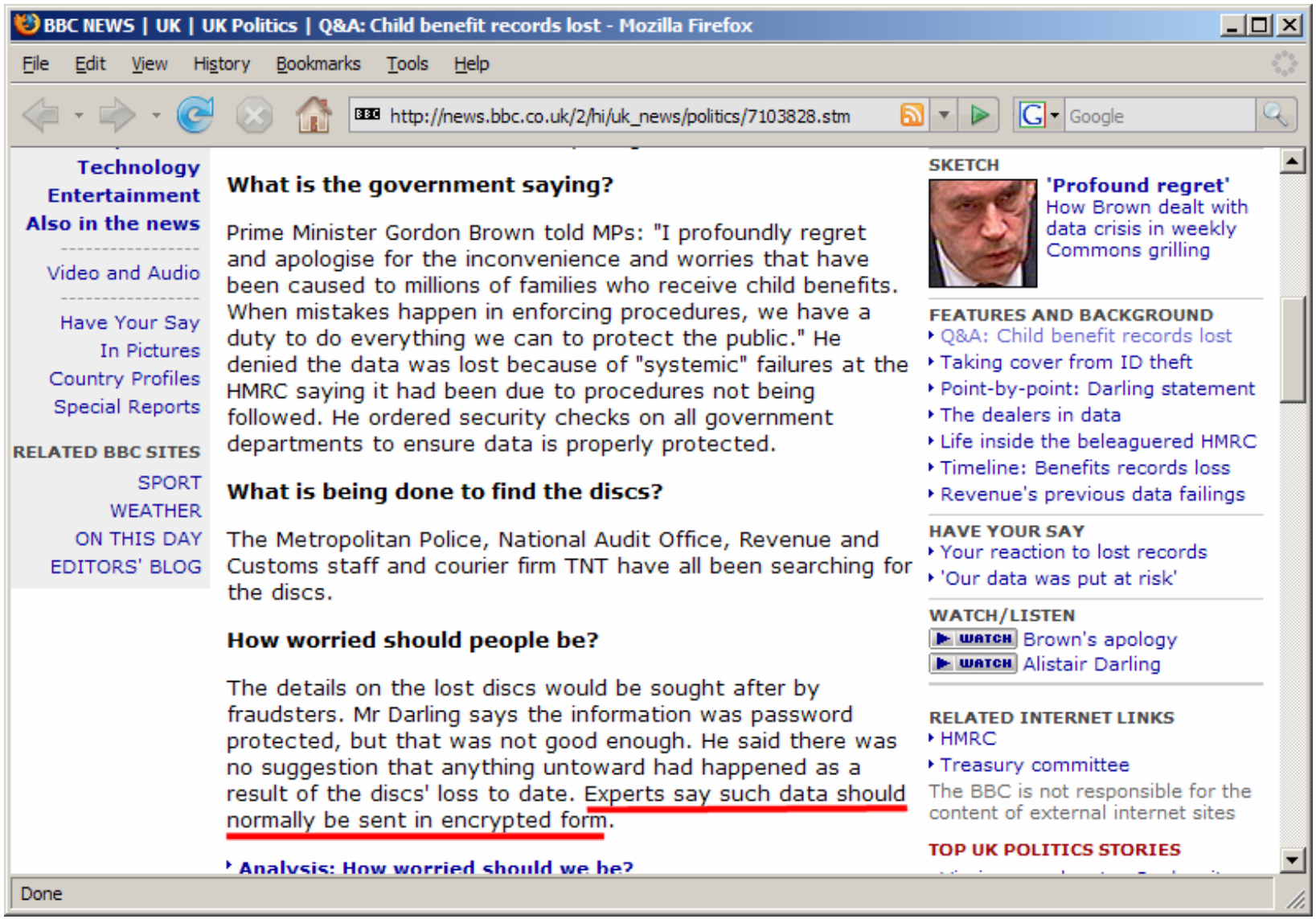
jschmidb@de.ibm.com

z/VSE und z/VM mit Linux on System z
GSE Frühjahrstagung
07.- 09. April 2008 in Bonn

Themen

- Thema Verschlüsselung in der Presse
- Lösungen für VSE
- Weitere Verschlüsselungstechnologien

Thema Verschlüsselung in der Presse



Lösungen im VSE für die Verschlüsselung von Daten

Lösungen für VSE

- **Sichere Übertragung von Daten**
  - SSL
  - Secure FTP
  - Secure Telnet
  - Connectors mit SSL
  - CICS Web Support mit SSL
  - POWER PNET mit SSL
- **Sicheres Speichern von Daten**
  - TS1120 seit z/VSE 3.1
  - *Neu:* Encryption Facility for z/VSE V1.1 mit z/VSE 4.1.1
  - Vendor Lösungen: z.B. Dr Crypto von CSI, Appliances wie z.B. mainstorconcept hier am Vendorenstand
- **Crypto APIs**
  - SSL und Crypto APIs von Connectivity Systems, Inc.
  - CryptLib von XPS Software GmbH mit zusätzlichen Algorithmen

Siehe hierzu PDFs auf der VSE Homepage: http://www.ibm.com/servers/eserver/zseries/zvse/documentation/security.html

Encryption Facility for z/VSE

- **Host-basiertes Batch-Tool**
- **Verschlüsselung von SAM files, VSAM files, und VSE Library members, aber auch von kompletten backups, unabhängig vom jeweiligen backup Prozess (IDCAMS, POFFLOAD, Vendor, …)**
- **Sehr ähnlich zum “Encryption Facility for z/OS”**
  - http://www.ibm.com/servers/eserver/zseries/zos/encryption_facility/
  - Kompatibel zum Encryption Facility for z/OS V1.1 und V1.2 (“System z Datenformat”)
- **Algorithmen: TDES und AES-128 (z9 und höher)**
- **Ausnutzung der System z Crypto Hardware**
  - Crypto Karten und CPACF
- **MWLC fähig**
- **Zwei Hauptfunktionen**
  - Password-basierte Verschlüsselung
  - Public-key basierte Verschlüsselung

Beziehung zum z/OS Encryption Facility

IBM Encryption Facility for z/VSE, 1.1
Program number: 5686-CF8-40
Runs on: System z10, z9 EC, z9 BC
zSeries 890 or 990

Requires: z/VSE 4.1 (with DY46717) or higher;

Optional Priced Feature

- Supports encrypting and decrypting of data at rest (tapes, disk)
- Supports either Public Key/Private keys or passwords to create highly-secure exchange between partners
- Use z/OS Java Client or Decryption Client for z/OS for data exchange with client systems or decryption on z/OS.
- Use zvse@de.ibm.com mailbox for questions about z/OS Java Client and Decryption Client for z/OS when used in relation with VSE.

IBM Encryption Facility for z/OS, 1.1
Program number: 5655-P97
Runs on: System z10, z9 EC, z9 BC
zSeries 900 or 990
zSeries 800 or 890
Requires: z/OS 1.4 or higher; z/OS.e 1.4 or higher

Java Client
Web Downloadable

Java technology-based code that allows client systems (z/OS and non-z/OS) to decrypt and encrypt data for exchange with z/OS and z/VSE systems.

Decryption Client for z/OS
Web Downloadable

Decryption only code designed to run on z/OS systems. Can process data encrypted and compressed or just encrypted by the Encryption Facility for z/OS or Encryption Facility for z/VSE.

Feature:
Encryption Services

Optional Priced Feature

- Supports encrypting and decrypting of data at rest (tapes, disk)
- Supports either Public Key/Private keys or passwords to create highly-secure exchange between partners

Feature:
DFSMSdss Encryption

Optional Priced Feature

TS1120 Überblick

Verschlüsselungsoption über JCL

- // ASSGN
- // KEKL

IP

SSL

Benötigt einen EKM zur Schlüsselverwaltung

- Java-fähige Plattform
- z/OS, AIX, Linux, Windows, HP, Sun, etc.

Hardware-basierte Verschlüsselung:

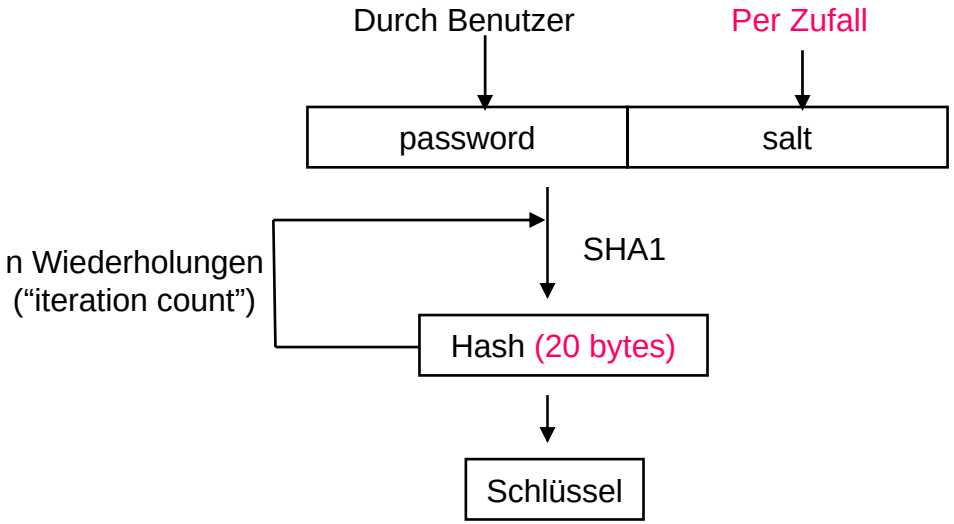
- Algorithmus: AES-256 fest eingestellt
- Public-Key basierte Verschlüsselung
- automatische Komprimierung vor der Verschlüsselung
- keine Host-Zyklen

Betriebsmodi des Encryption Facility

- **Passwort-basierte Verschlüsselung**
- **Public-Key basierte Verschlüsselung**

Passwort-basierte Verschlüsselung (PBE)

- **Beispiel einer einfachen sog. “Password-based key derivation function” (PBKDF1, beschrieben in RFC2898)**
- **Nachteil dieser Funktion:**
  - Erzeugter Schlüssel kann maximal so lang sein, wie die Ausgabe der verwendeten Hash Funktion, Bsp. MD5 = 16 bytes, SHA-1 = 20 bytes
- **Prozess:**



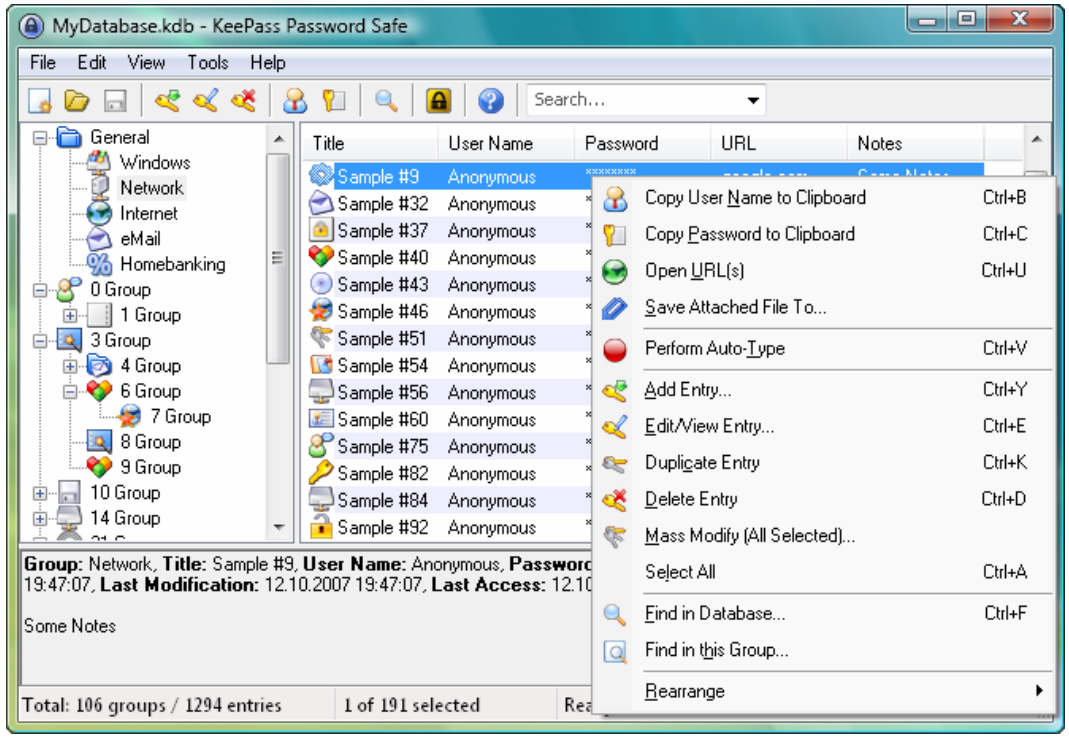
Ein damit verwandtes Verfahren, welches Schlüssel unbegrenzter Länge produzieren kann, wird im Encryption Facility verwendet.

Passwort-basierte Verschlüsselung (PBE)

- **Der encryption key (data key) wird generiert aus**
  - Dem geheimen Passwort
  - Einem “iteration count”, und
  - Einer 8-byte Zufallszahl (das sog. “salt”), welche für jeden Verschlüsselungsprozess verschieden ist.
- **Der “iteration count” und das “salt” werden im verschlüsselten Dataset abgelegt.**
  - D.h. diese zwei Werte sind nicht geheim, aber:
  - Wenn dieselben Daten ein zweites mal mit demselben Passwort und iteration count verschlüsselt werden, ergeben sich völlig unterschiedliche verschlüsselte Daten, weil das salt jedesmal anders ist.
- **Keine Notwendigkeit Schlüssel zu verwalten, aber**
- **Man muß seine Passwörter verwalten**
  - Hierzu gibt es verschiedene kommerzielle und auch freie Programme
  - Ein freies Programm: KeePass : http://keepass.sourceforge.net/

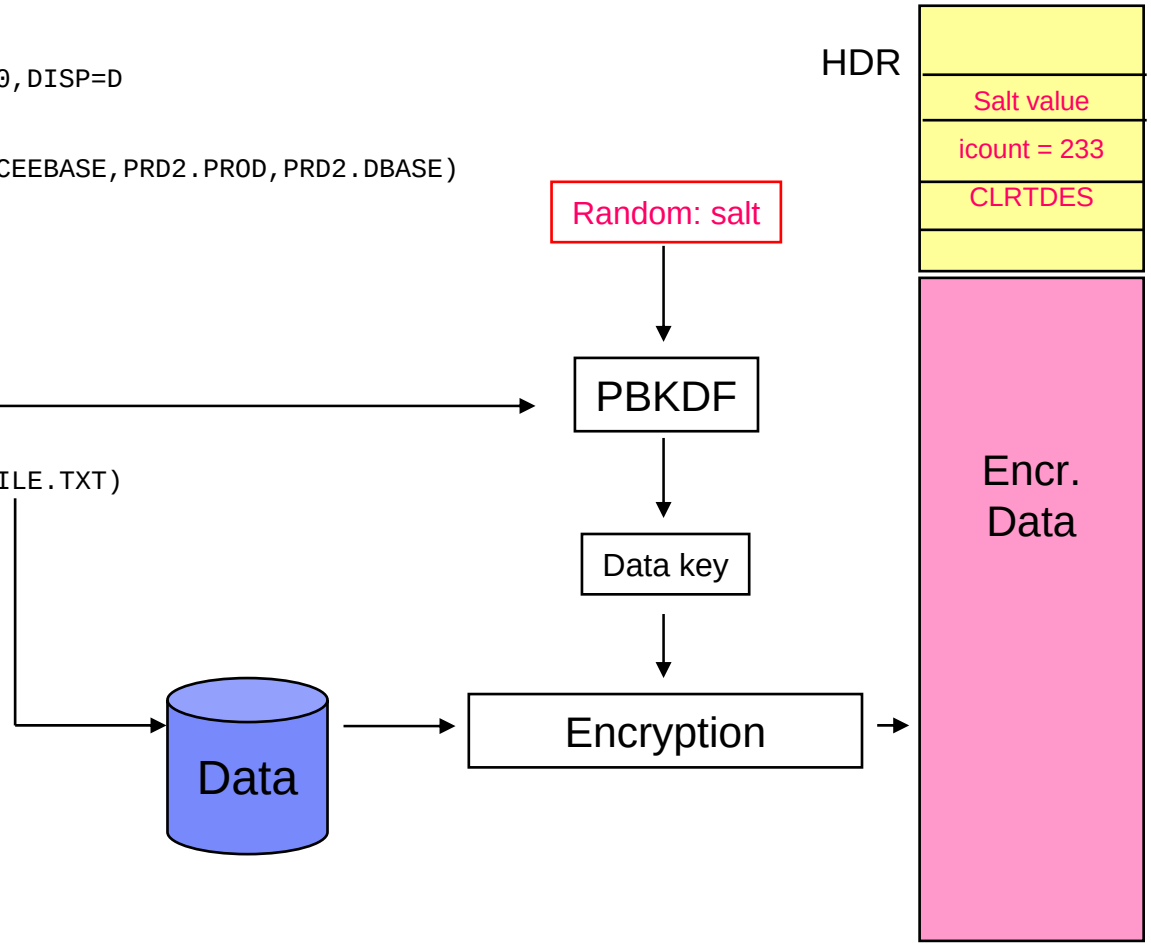
Keepass

- **Sicheres Aufbewahren von Passörtern**
- **Verschlüsselte Passwort-Datei**
- **Ein Master-Passwort erlaubt den Zugriff auf alle weiteren Passwörter**
- **Mögliche Verwendung mit VSE:**
  - Verwalten von Passwörtern für Encryption Facility
  - Ablegen der Keepass DB auf VSE, Doppelklick per Navigator



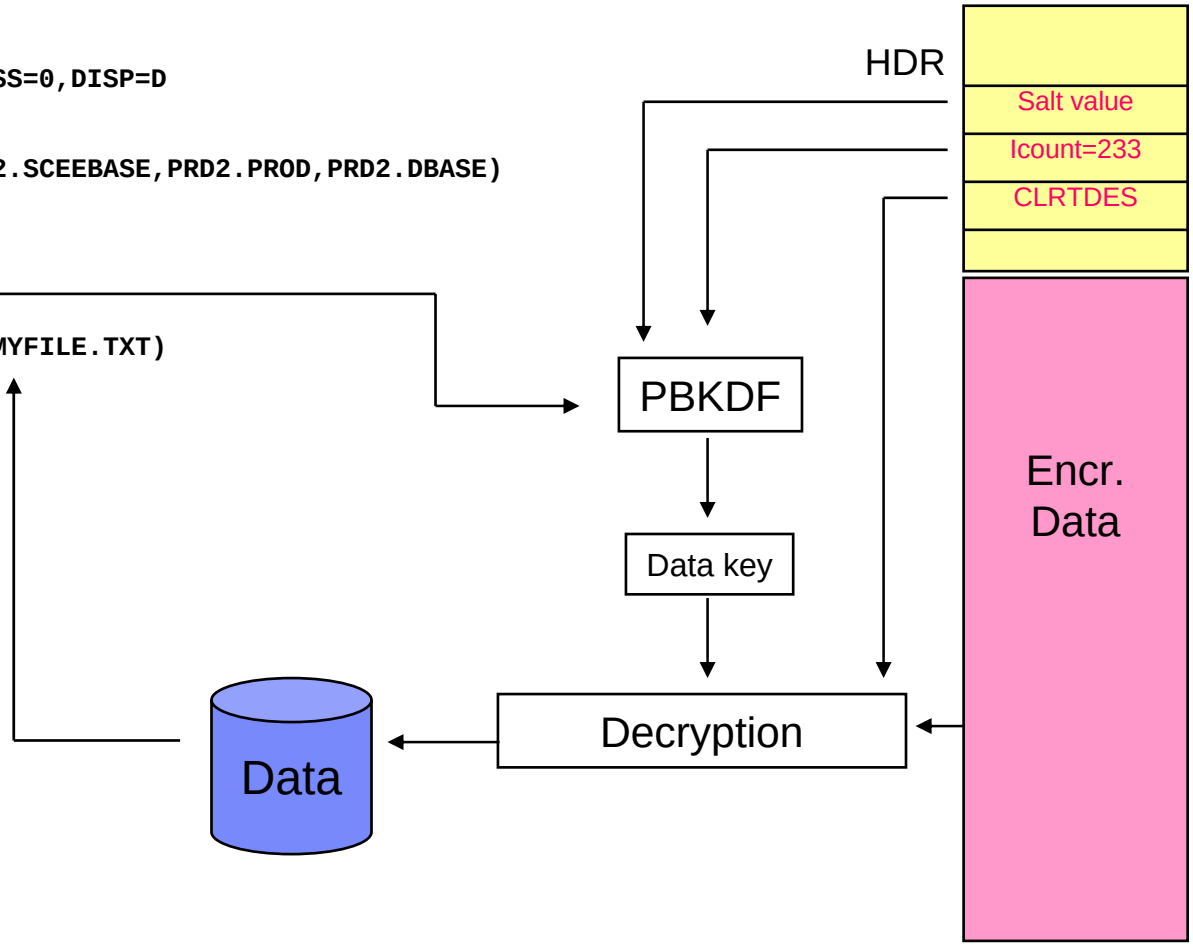
Job Beispiel für PBE Verschlüsselung

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



Job Beipiel für PBE Entschlüsselung

```
* $$ JOB JNM=DECMEM,CLASS=0,DISP=D
// JOB DECMEM
// LIBDEF *,SEARCH=(PRD2.SCEEBASE,PRD2.PROD,PRD2.DBASE)
// EXEC IJBEFVSE
DECRYPT
PASSWORD=BLAHBLAH
CLRFILE=DD:PRD2.CONFIG(MYFILE.TXT)
ENCFILE=DD:ENCDATA
/*
/&
* $$ EOJ
```

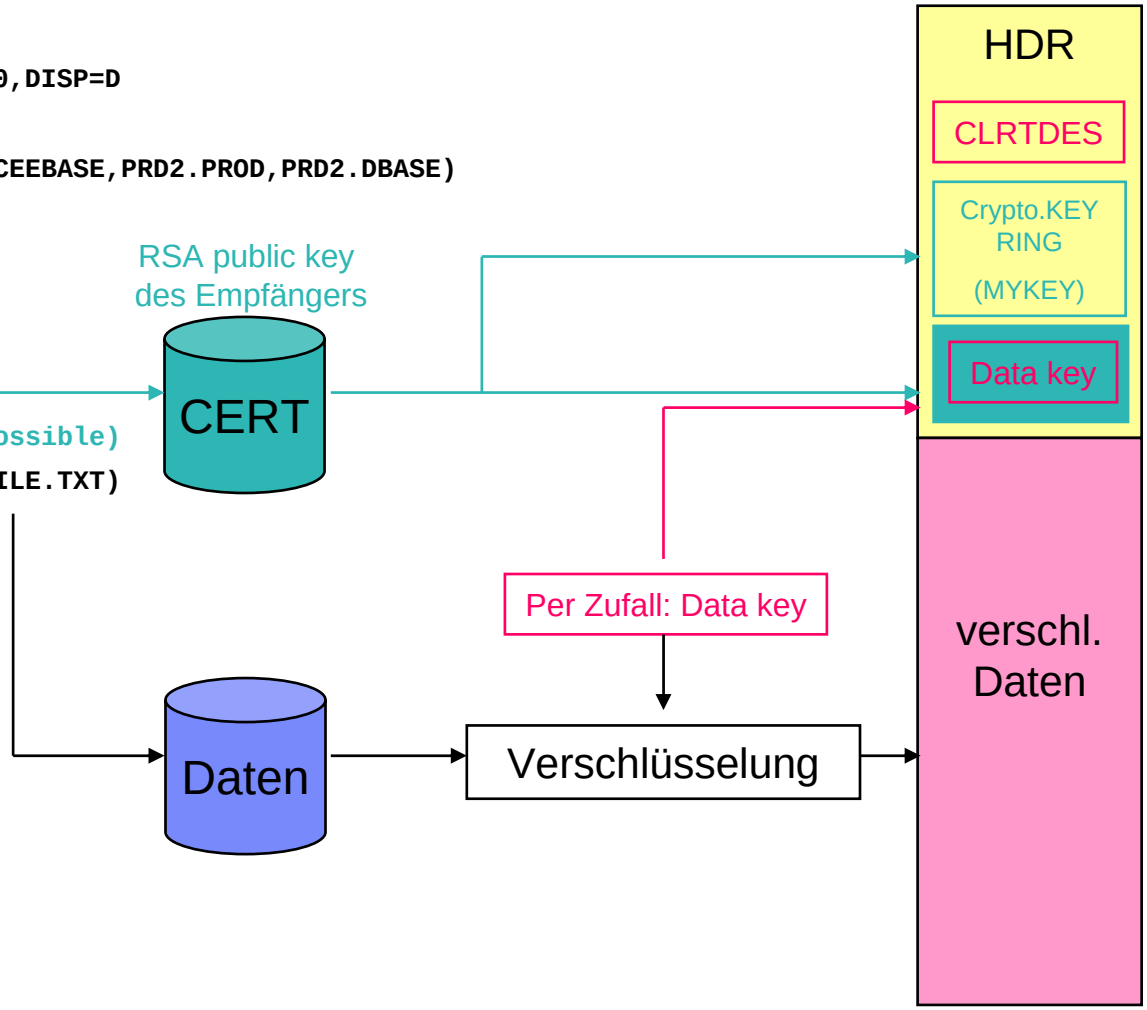


Public-Key basierte Verschlüsselung (PKE)

- Encryption key (data key) wird per Zufall generiert
- Data key wird dann mit dem public key des Empfängers der Daten verschlüsselt
- Data key wird zusammen mit den verschlüsselten Daten in das verschlüsselte dataset geschrieben
- Damit ist nur der Besitzer des zugehörigen privaten Schlüssels in der Lage den Data key zu entschlüsseln und damit wiederum die Daten zu entschlüsseln
- Es resultiert die Notwendigkeit, RSA Schlüsselpaare zu verwalten und auszutauschen
- Das kann z.B. mit dem Keyman/VSE Tool gemacht werden

Job Beispiel for public-key Verschlüsselung

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



Verfügbarkeit des Encryption Facility

- **Verfügbar seit 30.11.2007**
- **Voraussetzungen:**
  - CPU Assist Feature (CPACF), d.h. z890 / z990, z9, z10
  - z/VSE 4.1.1 refresh (APAR DY46717, PTF UD53196)
  - TCP/IP fix ZP15E214 für Public-key basierte Verschlüsselung
  - Crypto Express2 oder PCIXCC Karte für RSA Schlüssel der Länge 2048
- **Ist ein „Optional priced feature“**
- **Program number: 5686-CF8-40**
- **Dokumentation im z/VSE 4.1.1 Administration Buch, Kapitel 43**
  - Buch verfügbar auf CD-ROM, oder
  - Download als PDF von der VSE Homepage: http://www.ibm.com/servers/eserver/zseries/zvse/documentation/#vse

Positionierung zur TS1120

	TS1120	Encryption Facility
High volume backup/archiving	x	-
Verschlüsselung von Daten auf VSE Platten	-	x
Verschlüsselung von Daten zum anschließenden file transfer (z.B. FTP)	-	x
Lokales Archivieren von Daten	x	x
Austausch von Bändern mit Lokationen mit TS1120	x	-
Existierende Infrastruktur eines TS1120 mit EKM nutzen	x	-
Austausch von Daten mit Encryption Facility for z/OS V1.1	-	x
Austausch von Daten mit Non-Mainframe Plattformen (EF Java client)	-	x
Passwort-basierte Verschlüsselung	-	x
Public key basierte Verschlüsselung	x	x
Offload von CPU cycles	x	-

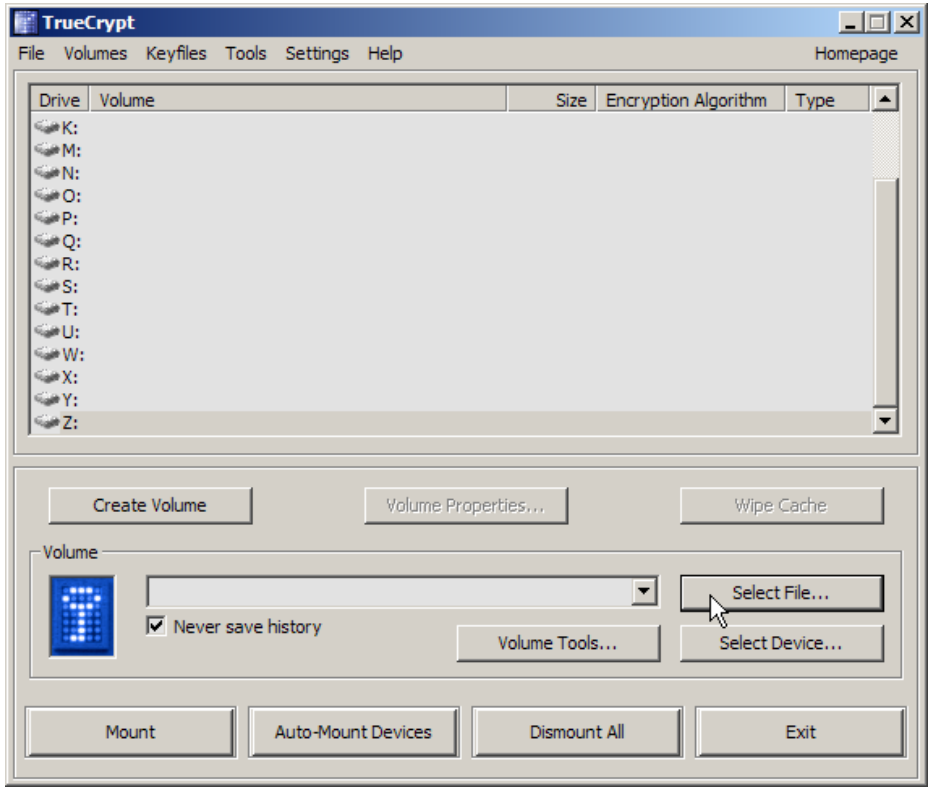
Migration von Bändern

- **Heutige Bänder haben ein Vielfaches der Kapazität von älteren Bändern.**
- **Deshalb bietet sich eine Migration von vielen kleinen Bändern auf wenige große an. Bei verschlüsselten Bändern ist zu beachten:**
- **TS1120:**
  - Die TS1120 erlaubt nur einen Schlüssel pro Band.
  - Dieser ist physikalisch auf der Bandkassette gespeichert.
  - „Umkopieren“ bedeutet damit immer auch eine Neuverschlüsselung
- **Encryption Facility**
  - Jedes Dataset ist mit eigenem Password bzw. Public-Key verschlüsselt, nicht an HW gebunden.
  - Mehrere verschlüsselte Datasets können auf ein Band kopiert werden und von dort per TLBL wieder separat gelesen und entschlüsselt werden (allerdings nicht VSAM Datasets wegen interner Tape marks)

Weitere Verschlüsselungstechnologien über das VSE hinaus

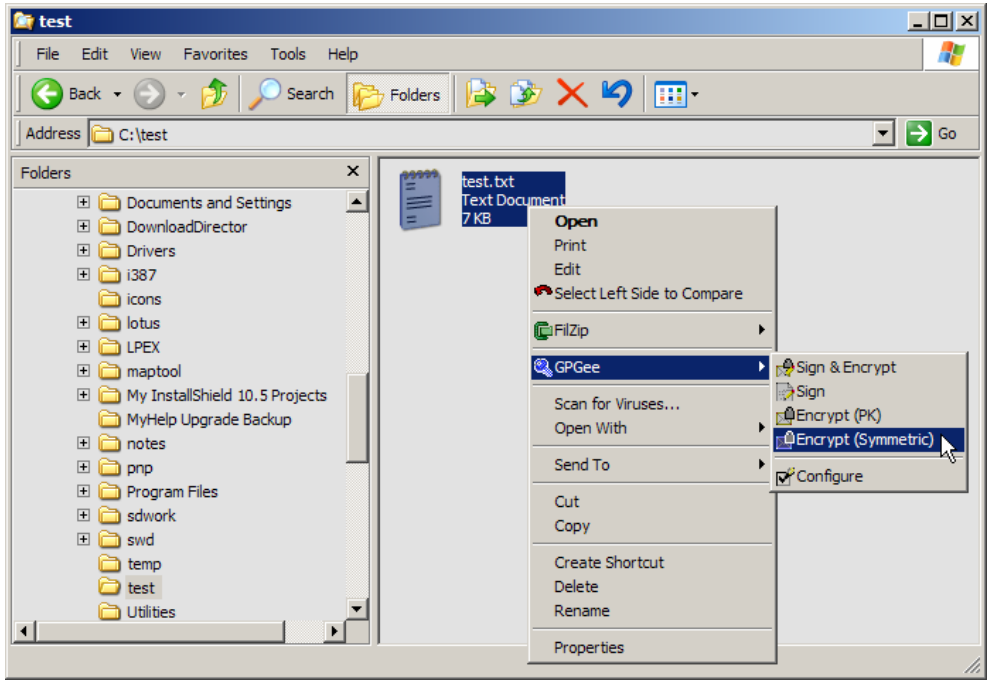
Truecrypt

- **Freie Open-Source Software**
- **Verschlüsselte Dateien werden in der Art eines Laufwerks vorformatiert und auch als Laufwerk zugegriffen**
- **Mögliche Verwendung mit VSE:**
  - Backups auf VTAPE und die AWSTAPE Dateien liegen in einem Truecrypt Volume in einem Windows oder Linux



Pretty-good privacy (PGP)

- **Entwickelt 1991 von Phil Zimmermann**
- **Definiert in RFCs 2440 / 4880**
- **Basiert auf dem „web-of-trust“ Modell, im Gegensatz zum herkömmlichen hierarchischen Trust-Modell**
- **Mögliche Verwendung mit VSE:**
  - Nachträgliche Verschlüsselung von aus dem VSE transferierten Daten
- **Downloads:**
  - http://www.gnupg.org
  - http://www.gpg4win.org



Webseiten

- **PDFs auf VSE Homepage: How to setup SecureFTP, SecureTelnet, CICS Web Support, etc.**
  http://www.ibm.com/servers/eserver/zseries/zvse/documentation/security.html
- **Keyman/VSE tool und 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/
- **Encryption Facility for z/VSE, Dokumentation im VSE Administration Buch**
  http://www.ibm.com/servers/eserver/zseries/zvse/documentation/#vse
- **IBM Encryption Facility for z/OS Java Client**
  http://www.ibm.com/servers/eserver/zseries/zos/downloads/#efclient
- **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
- **Truecrypt**
  http://www.truecrypt.org/
- **KeePass Password Safe – a free Open Source Password Manager for many operating systems**
  http://keepass.sourceforge.net/
- **PGP**
  http://www.gpg.org/
  http://www.gpg4win.org

Fragen