Kassenärztliche Bundesvereinigung

Körperschaft des öffentlichen Rechts

IT in der Arztpraxis

Handbuch KBV-Kryptomodul XKM

[KBV_ITA_AHEX_Handbuch_Kryptomodul]

Dezernat Digitalisierung und IT

10623 Berlin, Herbert-Lewin-Platz 2

Kassenärztliche Bundesvereinigung

Version	1.35
Datum:	14.02.2023
Kennzeichnung:	Öffentlich
Status:	In Kraft

DOKUMENTENHISTORIE

Version	Datum	Autor	Änderung	Begründung	Seite
1.35	14.02.2023	KBV	Schlüssel für QSKE und QSHLT aktualisiert	Schlüsselpaar ausgetauscht	12
1.34	14.02.2022	KBV	Schlüssel für KV, EHKS, DA, ADMP und ZervixZyto aktualisiert	Schlüsselpaar ausgetauscht	12
1.33	13.11.2020	KBV	Java Version		8
1.32	06.03.2019	KBV	Überarbeitung Systemvoraussetzungen Aktualisierung des Aufrufs von der Kommandozeile im Beispiel textuelle Anpassungen bzgl. des geänderten Dateiformates	Es wird mindestens Java 8 erwartet	8 23 alle
1.31	09.08.2018	KBV	Anpassung bzgl. der neuen Schlüssel für QSKE und QSHLT		12
1.30	15.05.2018	KBV	Anpassung der Indexe		12
1.29	29.01.2018	KBV	Aktualisierung der Schlüssel für DA, ADMP, KVKol, KV, Dialyse, ZervixZyto, EHKS Streichung der LDT 2-Unterstützung Streichung der Disketten Unterstützung	Vorgesehener zweijähriger Schlüsseltausch	12 8 12 27 10 14 15 17 21
1.28	02.05.2017	KBV	Versionsnummer der Schlüssel QSMG, QSHGV und QSHGVK aktualisiert	Schlüsselpaar ausgetauscht	12
1.27	08.08.2016	KBV	Versionsnummern der QSKE-Schlüssel erhöht Neuen Arbeitsmodus QSHLT hinzugefügt Abschnit 7.1 (Netzwerkinstallation) überarbeitet	Schlüsselpaar QSKE ausgetauscht neuer Schlüssel QSHLT XKM ist nicht multi-user-fähig	12 30
1.26	09.02.2016	KBV	Aktualisierung von Schlüsselnamen	Austausch der Schlüsselpaare	12
1.25	05.05.2015	KBV	Aktualisierung eines Schlüsselnamens	Austausch des Schlüssels QSHGVK	12
1.24	10.02.2015	KBV	Aktualisierung der Schlüsselnamen	Austausch der Schlüsselpaare QSHGV und QSMG	12
1.23	12.11.2014	KBV	Schlüssel QSKE hinzugefügt	Neue QS-Dokumentation Kapselendoskopie	12
1.22	14.05.2014	KBV	Dokument **redaktionell** bzgl. veralteter Links überarbeitet		
1.21	03.02.2014	KBV	Aktualisierung der Schlüsselnamen	Austausch der Schlüsselpaare	11

1.20	11.02.2013	KBV	Verweise auf Registerstruktur entfernt. Neue Arbeitsmodi	Ablösung der Register -Registerstruktur	11 27
1.19	20.07.2012	KBV	Redaktionelle Anpassungen		
1.18	23.01.2012	KBV	Beschreibung Returnwerte und Lösungsmöglichkeiten		10
1.17	19.01.2012	KBV	Neue Kapitel zu Schlüssel, Schlüsselverwaltung, Konfigurations- und Handhabungsempfehlungen	Zusammenführung mit Dokument XKM-Anwenderhinweise	24 28 30

INHALTSVERZEICHNIS

DOKUMENTENHISTORIE 2

INHALTSVERZEICHNIS 4

ABBILDUNGSVERZEICHNIS 6

TABELLENVERZEICHNIS 7

1 EINLEITUNG 8

1.1 Produktleistung 8

1.2 Systemvoraussetzungen 8

1.3 Nutzungsbedingungen 8

2 VERZEICHNISSTRUKTUR 9

3 ARBEITEN MIT DEM XKM 10

3.1 Allgemeine Handhabung 10

3.2 Die Arbeitsmodi 11

3.2.1 Vordefinierte Arbeitsmodi 11

3.2.2 Benutzerdefinierte Arbeitsmodi 13

3.3 Die Konfigurationsdatei 14

3.3.1 Konfigurationsschalter 15

3.4 Arbeiten im Kommandozeilenmodus 16

3.4.1 Kommandozeilenparameter 17

3.5 Umgebungsvariablen 18

3.6 Arbeiten mit grafischer Oberfläche 19

3.6.1 Der Standard-Dialog 19

3.7 Arbeiten mit Wechselmedien (CD, ...) 21

3.7.1 Notwendige Schritte zur Integration eines Wechselmediums 21

3.8 Hilfsprogramme des XKMs 22

3.8.1 Erzeugen benutzerspezifischer Schlüsselpaare 22

3.8.2 Erzeugen allgemeiner Schlüsselpaare 22

4 STARTEN DES XKM 23

4.1 Starten des XKM von Konsole/Desktop 23

5 SCHLÜSSEL 24

5.1 Individuelles Benutzerschlüsselpaar ... 24

5.2 Privater Schlüssel für zertifizierte Systemanbieter ... 26

5.3 Schlüssel für die Standardkommunikationsszenarien den Praxen ... 27

6 SCHLÜSSELVERWALTUNG / ARBEITSMODI 28

6.1 Einbinden von Schlüsseln ... 28

6.2 Testen von Ver- und Entschlüsselung ... 29

7 KONFIGURATIONS- UND HANDHABUNGSEMPFEHLUNGEN 30

7.1 Backup der Schlüssel und der Schlüsselzuordnung ... 30

7.2 GUI-Optionen für Endanwender sperren ... 30

7.3 XKM-Installation auf einem Netzlaufwerk ... 30

7.4 Starten des XKM aus einem Java-Programm heraus ... 31

8 DAS VERSCHLÜSSELUNGSVERFAHREN DES XKM 32

9 AUFBAU DES XKM-HEADER 33

ABBILDUNGSVERZEICHNIS

Abbildung 1: Der Standarddialog 19
Abbildung 2: XKM Kryptomodul im Konfigurations-Dialog 20
Abbildung 3: Protokoll Test-Ver-/Entschlüsselung 29
Abbildung 4: Konsole Test-Ver-/Entschlüsselung 29

TABELLENVERZEICHNIS

Tabelle 1: Übersicht der XKM Ordner 9
Tabelle 2: Aufrufmöglichkeiten des XKM 10
Tabelle 3: Liste der Returnwerte der JAVA Applikation 11
Tabelle 4: Verwendete Schlüssel des Kryptomoduls 12
Tabelle 5: Konfigurationsschalter 16
Tabelle 6: Kommandozeilenparameter 17
Tabelle 7: Aufruf zum Erzeugen benutzerspezifischer Schlüsselpaare 22
Tabelle 8: Aufruf zum Erzeugen allgemeiner Schlüsselpaare 22
Tabelle 9: Strukturbeschreibung des XKM Headers 33

1 Einleitung

Dieses Dokument gibt einen kurzen Überblick über die Arbeitsweise und eine genaue Beschreibung zur Installation und Ausführung des KBV-Kryptomoduls XKM.

Das XKM ist auf allen Computersystemen lauffähig, für die die Java Laufzeitumgebung in der Version 11 oder höher verfügbar ist.

1.1 Produktleistung

Das XKM dient der Verschlüsselung sensibler Daten, die mittels Speichermedium oder Leitung zwischen Arztpraxis und Datenannahmestelle oder zur KV übermittelt werden. Abhängig von der Berechtigung können die verschlüsselten Daten vom XKM wieder entschlüsselt werden. Bei der Datenübermittlung mittels Datenträger unterstützt das XKM das Aufsplitten der Daten, falls die Kapazität eines einzelnen Datenträgers nicht ausreicht.

1.2 Systemvoraussetzungen

Benötigt wird ein Computersystem, das leistungsmäßig mit einem IBM-kompatiblen PC, 256 MB Hauptspeicher und einem Pentium-Prozessor 500Mhz oder höher vergleichbar ist.

Langsamere Prozessoren und weniger Hauptspeicher erhöhen stark die Laufzeit des Programms. Das XKM benötigt ca. 50 MByte Festplattenplatz.

Es wird die Java Laufzeitumgebung 11 oder höher benötigt. Das XKM ist auch lauffähig unter der aktuellste LTS-Version Java 17 (OpenJDK oder OracleJDK).

1.3 Nutzungsbedingungen

Die KBV übernimmt den Support bei der Verwendung des XKMs in den aufgelisteten Arbeitsmodi im Kapitel 3.2.1. Die KBV übernimmt keinen Support bei selbstdefinierten Arbeitsmodi und bei selbstdefinierten Schlüsseln.

2 Verzeichnisstruktur

Im Installationsverzeichnis befinden sich Batchdateien bzw. Shellskripte, die das KBV-Kryptomodul mit unterschiedlichen Optionen bzw. Konfigurationsdateien ausführen.

Die Verzeichnisstruktur des KBV-Kryptomoduls hat den folgenden Aufbau:

Ordner	Funktion des Ordners
'Ausschuss'	Dieser Ordner dient als Ablage für alle Dateien, die nicht entschlüsselt werden konnten.
'Bearbeitet'	Dieser Ordner wird nur im Bedarfsfall vom XKM erstellt. Hier werden Dateien abgelegt, die im Serverbetrieb ohne Löschmodus (-l) verarbeitet wurden.
'Bin'	Dieser Ordner beinhaltet alle Java-Archive, die zur Ausführung des KBV-Kryptomoduls benötigt werden.
'Doku'	Dieser Ordner enthält Dokumentationen zum XKM. Zum Beispiel dieses Dokument.
'Entschluesselt'	Dieser Ordner enthält alle Dateien die entschlüsselt wurden.
'Konfig'	Dieser Ordner enthält die Konfigurationsdateien ’config.xml’’ und ’schluessel.xml’
'Listen'	Dieser Ordner dient als Default-Ablage für die Protokolle.
'Quelle'	Dieser Ordner ist das Eingangsverzeichnis für die zu verschlüsselnden Dateien.
'Schluessel'	Dieser Ordner enthält den konfigurierbaren Teil der verwendeten Schlüssel.
'System'	Wird ausschließlich für interne Abläufe benötigt.
'Verschluesselt'	Dieser Ordner dient als Ablage für die verschlüsselten Dateien. Zugleich ist der Ordner auch das Eingangsverzeichnis beim Entschlüsseln von Dateien.

Tabelle 1: Übersicht der XKM Ordner

3 Arbeiten mit dem XKM

3.1 Allgemeine Handhabung

Der Aufruf des XKM-Hauptprogramms erfolgt über die im Installationsverzeichnis mitgelieferten Batch- bzw. Skriptdateien:

Dateiname	Funktion
StartKryptomodul.bat StartKryptomodul.sh	Aufruf des XKM im Konsolenbetrieb. Die Steuerung des Kryptomoduls erfolgt mittels Konfigurationsdatei und Übergabeparameter.
StartGUIKryptomodul.bat StartGUIKryptomodul.sh	Aufruf des XKM im Betrieb mit grafischer Oberfläche. Die Steuerung des Kryptomoduls erfolgt mittels Konfigurationsdatei und Übergabeparameter.

Tabelle 2: Aufrufmöglichkeiten des XKM

Zu jedem Ver- bzw. Entschlüsselungsvorgang wird im *Listen*-Verzeichnis eine Protokolldatei erzeugt, die einen kurzen Statusbericht zum Ergebnis des Vorgangs enthält. Der Name der Protokolldatei setzt sich aus dem Namen der bearbeiteten Datei und einem angehängten Suffix zusammen, das sich aus dem gewählten Ausgabeformat (s. Kapitel 3.3.1, Schalter *Protokollformat*) ergibt.

Weiterhin wird der Status des Verarbeitungsvorgangs über den Returnwert der Java-Applikation an den Aufrufer zurückgegeben. Nachfolgend eine Liste der möglichen Returnwerte:

Returnwert		Lösungsmöglichkeit
0:	ok	
1:	Unerwartetes Folgepaket	Zu einer ungesplitteten Datenlieferung wurden unerwartete Folgepakete gefunden. Prüfen Sie den Dateneingang und setzen Sie sich ggfls. mit dem Absender in Verbindung.
2:	Keine Datei bearbeitet	Prüfen Sie, ob im Eingangsverzeichnis bearbeitbare Dateien vorhanden sind.
3:	Konfigurationsfehler	Das XKM kann nicht starten, da ein Konfigurationsfehler vorliegt Beachten Sie den zusätzlichen XKM-Fehlertext.
4:	interner Fehler	Programmfehler oder nicht näher bestimmbarer Fehler. Setzen Sie sich mit unserem Support in Verbindung.
5:	Verarbeitung abgebrochen	(Manueller Abbruch des Programms durch Benutzer.)
6:	Schlüsselfehler	Es wurde der falsche Schlüssel im XKM eingebunden. Prüfen Sie ob der aktuellste Schlüssel verwendet wird. Fragen Sie auf der Absenderseite nach, mit welchem Schlüssel die Daten verschlüsselt wurden. In seltenen Fällen kann auch eine auf dem Transportweg beschädigte Datei zu diesem Fehler führen.
7:	Hardwareproblem	Datei kann nicht verarbeitet (lesen, schreiben, löschen) werden: a) Prüfen Sie, ob die notwendigen Zugriffsrechte auf die Datei oder das Verzeichnis bestehen. b) Prüfen Sie, ob ein anderes Programm die Datei in Bearbeitung hat und somit blockiert.
8:	unbekannter Endestatus	(Wird nicht mehr verwendet)

Returnwert		Lösungsmöglichkeit
9:	kein Folgepaket gefunden	Zusammenfassung gesplitteter Dateien ist nicht möglich, weil eine oder mehrere Dateien fehlen. Stellen Sie sicher, dass alle zusammengehörigen Split-Dateien übermittelt wurden
10:	Protokollierungsproblem	Die Protokoll-Datei kann nicht erstellt werden. Eine namensgleiche Datei ist bereits durch einen PDF-Reader oder ein anderes Programm geöffnet.
11:	Fehler bei Prüfsummenbildung	Ein Prüfsummencheck hat ergeben, dass die verschlüsselten Daten verändert wurden. Stellen Sie sicher, dass die Dateien auf dem Transportweg korrekt übertragen werden.

Tabelle 3: Liste der Returnwerte der JAVA Applikation

3.2 Die Arbeitsmodi

3.2.1 Vordefinierte Arbeitsmodi

Das XKM unterscheidet zwischen der Ver- und der Entschlüsselung. In der allgemeinsten Form werden diese Vorgänge durch die beiden Arbeitsmodi *Verschluesselung* und *Entschluesselung* repräsentiert. Zusätzlich zu diesen beiden Modi gibt es eine Reihe weiterer vordefinierter Arbeitsmodi, die bestimmte Aktivitäten wie Dateifilter oder das automatische Anhängen einer Datei (z.B. Kommunikationssatz) erlauben. Daneben werden diesen Modi unterschiedlichen Schlüsseldateien zugeordnet.

Die Versionen der Schüssel in der folgenden Tabelle listen die zum Zeitpunkt der Veröffentlichung dieses Dokuments aktuell-gültigen Schlüssel. Diese werden ebenfalls auf der ITA-Webseite veröffentlicht.

Achtung: Die in der Tabelle verwendete Versionsnummer des Schlüssels muss nicht der aktuell gültigen Version entsprechen. Bitte entnehmen Sie die aktuellen (öffentlichen) Schlüssel immer direkt der ITA-Webseite.

Name des Arbeitsmodus	Dateifilter		Schlüsseldatei	Index	Besonderheit
	Eingang	Ausgang			
Verschluesselung	*	*.XKM	Je nach Benutzerkennung		-
Entschluesselung	*.XKM	*	Je nach Benutzerkennung		-
DMP_Verschluesselung	*.zip	*.zip.XKM	Oeffentlich_DA_V08.pub	3	-
DMP_Entschluesselung	*.zip.XKM	*.zip	Privat_DA_V08.pfx	4	-
Koloskopie_Verschluesselung	*.zip	*.zip.XKM	Oeffentlich_KVKol_V06.pub	5	-
Koloskopie_Entschluesselung	*.zip.XKM	*.zip	Privat_KVKol_V06.pfx	6	-
Abrechnungs_Verschluesselung	*.con	*.con.XKM	Oeffentlich_KV_V08.pub	7	Komusatz wird unterstützt
Abrechnungs_Entschluesselung	*.con.XKM	*.con	Privat_KV_V08.pfx	8	-
Dialyse_Verschluesselung	*.zip	*.zip.XKM	Oeffentlich_Dialyse_V06.pub	9	-
Dialyse_Entschluesselung	*.zip.XKM	*.zip	Privat_Dialyse_V06.pfx	10	-
ADMP_Verschluesselung	*	*.XKM	Oeffentlich_ADMP_V08.pub	11	-
ADMP_Entschluesselung	*.XKM	*	Privat_ADMP_V08.pfx	12	-
ZervixZyto_Verschluesselung	*.zip	*.zip.XKM	Oeffentlich_ZervixZyto_V08.pub	13	-
ZervixZyto_Entschluesselung	*.zip.XKM	*.zip	Privat_ZervixZyto_V08.pfx	14	-
TEST_Verschluesselung	*	*.XKM	Oeffent-lich_Testschluessel_V01.pub	15	zu Testzwecken
TEST_Entschluesselung	*.XKM	*	Privat_Testschluessel_V01.pfx	16	zu Testzwecken
EHKS_Verschluesselung	*.zip	*.zip.XKM	Oeffentlich_EHKS_V08.pub	17	-
EHKS_Entschluesselung	*.zip.XKM	*.zip	Privat_EHKS_V08.pfx	18	-
ApothekenRZ_Verschluesselung	*	*.XKM	Oeffent-lich_ApothekenRZ_V01.pub	19	-
ApothekenRZ_Entschluesselung	*.XKM	*	Privat_ApothekenRZ_V01.pfx	20	-
WTK_Verschluesselung	*.wtk	*.wtk.XKM	Oeffentlich_KV_V08.pub	21	Komusatz wird unterstützt
WTK_Entschluesselung	*.wtk.XKM	*.wtk	Privat_KV_V08.pfx	22	-
VIK_Verschluesselung	*.vik	*.vik.XKM	Oeffentlich_KV_V08.pub	23	Komusatz wird unterstützt
VIK_Entschluesselung	*.vik.XKM	*.vik	Privat_KV_V08.pfx	24	-
QSHGV_Verschluesselung	*.zip	*.XKM	Oeffentlich_QSHGV_V05.pub	25	-
QSHGV_Entschluesselung	*.zip.XKM	*.zip	Privat_QSHGV_V05.pfx	26	-
QSMG_Verschluesselung	*.zip	*.XKM	Oeffentlich_QSMG_V05.pub	27	-
QSMG_Entschluesselung	*.zip.XKM	*.zip	Privat_QSMG_V05.pfx	28	-
QSHGVK_Verschluesselung	*.zip	*	Oeffentlich_QSHGVK_V05.pub	29	-
QSHGVK_Entschluesselung	*.zip.XKM	*.zip	Privat_QSHGVK_V05.pfx	30	-
QSKE_Verschluesselung	*.zip	*	Oeffentlich_QSKE_V05.pub	31	-
QSKE_Entschluesselung	*.zip.XKM	*.zip	Privat_QSKE_V05.pfx	32	-
QSHLT_Verschluesselung	*.zip	*	Oeffentlich_QSHLT_V04.pub	33	-
QSHLT_Entschluesselung	*.zip.XKM	*.zip	Privat_QSHLT_V04.pfx	34	-

Tabelle 4: Verwendete Schlüssel des Kryptomoduls

Erläuterung zum Dateifilter:
Der Dateifilter des Modus ‚Verschluesselung' übernimmt als Input eine Datei beliebigen Namens (Platzhalter '*') und hängt nach erfolgter Verschlüsselung das Suffix ‚.XKM' an. Bei der Entschlüsselung wird das angehängte Suffix wieder entfernt, so dass der ursprüngliche Dateiname wiederhergestellt wird.

Bei der ‚DMP_Verschluesselung' werden dagegen nur Dateien mit der Endung ‚.zip' akzeptiert, die dann in Dateien mit der Endung ‚.zip.XKM' umgewandelt werden. Bei der ‚DMP_Entschluesselung' wird der ursprüngliche Dateiname wiederhergestellt.

Dito für die anderen Arbeitsmodi.

Ab XKM Version 1.14 gibt es die Möglichkeit, bei Entschlüsselungen in vordefinierten Arbeitsmodi mit mehreren Schlüsseln zu arbeiten. Hierzu können im gewünschten Arbeitsmodus einfach weitere Schlüssel benannt werden. Das XKM geht die Liste der spezifizierten Schlüssel durch, bis eine erfolgreiche Entschlüsselung möglich ist:

```
<arbeitsmodus>
        <index>4</index>
        <partnerindex>3</partnerindex>
        <typ>entschluesselung</typ>
        <name>DMP_Entschluesselung</name>
        <schluessel>Privat_Schluessel1.pfx</schluessel>
        <schluessel>Privat_Schluessel2.pfx</schluessel>
        <suffix>null</suffix>
        <systempfad>true</systempfad>
        <benutzermodus>false</benutzermodus>
</arbeitsmodus>
```

3.2.2 Benutzerdefinierte Arbeitsmodi

Benutzerdefinierte Arbeitsmodi werden in der XML-Datei *schluessel.xml* beschrieben. Ein beispielhafter Inhalt für je einen Verschlüsselungs- und einen Entschlüsselungsmodus:

```
<arbeitsmodi>
   <arbeitsmodus>
      <typ>verschluesselung</typ>
      <name>Meine_Verschluesselung</name>
      <schluessel>Mein_oeffentlicher_Schluessel.pub</schluessel>
   </arbeitsmodus>

   <arbeitsmodus>
      <typ>entschluesselung</typ>
      <name>Meine_Entschluesselung</name>
      <schluessel>Mein_privater_Schluessel.pfx</schluessel>
   </arbeitsmodus>
</arbeitsmodi>
```

3.3 Die Konfigurationsdatei

Die Steuerung des XKMs erfolgt mit Hilfe einer Konfigurationsdatei, die defaultmäßig *Konfig/config.xml* heißt. Eine Konfigurationsdatei mit diversen Vorbelegungen befindet sich nach der Installation im Unterverzeichnis 'Konfig'.

Die Pfadangaben in der Konfigurationsdatei müssen eventuell dem jeweiligen Betriebssystem angepasst werden. Die Konfigurationsdateien im Lieferumfang sind so voreingestellt, dass keinerlei Anpassungen nötig sind. Als Trennzeichen für Verzeichnisse wird das Zeichen „/" verwendet. Diese Voreinstellung erlaubt die Nutzung gleicher Konfigurationsdateien auf verschiedenen Betriebssystemen (Windows, Unix, Linux, ...). Relative Pfadangaben werden als relativ zum Installationsverzeichnis betrachtet.

Der Inhalt einer kompletten *config.xml*-Datei könnte zum Beispiel wie folgt aussehen:

```
<Konfiguration>
   <quelle>Quelle/</quelle>
   <ausschuss>ausschuss/</ausschuss>
   <entschluesselt>entschluesselt/</entschluesselt>
   <verschluesselt>verschluesselt/</verschluesselt>
   <arbeitsmodus>Verschluesselung</arbeitsmodus>
   <protokolldatei>Listen/Protokoll</protokolldatei>
   <system>System</system>
   <schluesselpfad>Schluessel/</schluesselpfad>
   <protokollformat>PDF</protokollformat>
   <paketgroesse>unbegrenzt</paketgroesse>
   <!-- <pruefinfo/>-->
   <!-- <konfigdialog/>-->
</Konfiguration>
```

3.3.1 Konfigurationsschalter

Schalter	Zulässige Werte	Funktion
Quelle	Pfadangabe	Dateneingangsverzeichnis bei der Verschlüsselung
Verschluesselt	Pfadangabe	In diesem Verzeichnis werden die verschlüsselten Dateien abgelegt. Ist gleichzeitig Dateneingangsverzeichnis bei der Entschlüsselung.
Entschluesselt	Pfadangabe	In diesem Verzeichnis werden die entschlüsselten Dateien abgelegt.
Ausschuss	Pfadangabe	Ablage von Dateien, die nicht entschlüsselt werden konnten.
Arbeitsmodus	Verschluesselung Entschluesselung DMP_Verschluesselung DMP_Entschluesselung Abrechnungs_Verschluesselung Abrechnungs_Entschluesselung ADMP_Verschluesselung ADMP_Entschluesselung ZervixZyto_Verschluesselung ZervixZyto_Entschluesselung TEST_Verschluesselung TEST_Entschluesselung EHKS_Verschluesselung EHKS_Entschluesselung ApothekenRZ_Verschluesselung ApothekenRZ_Entschluesselung WTK_Verschluesselung WTK_Entschluesselung QSHGV_Verschluesselung QSHGV_Entschluesselung QSHGVK_Verschluesselung QSHGVK_Entschluesselung QSMG_Verschluesselung QSMG_Entschluesselung	Festlegung des Arbeitsmodus Neben den bereits vordefinierten Modi können hier auch benutzerdefinierte Arbeitsmodi angegeben werden.
Protokolldatei	Dateiangabe	Name der Datei für die Protokollausgabe
Protokollformat	Unterstützte Formate: PDF, TEXT, HTML, XLS, RTF, XML, JRPRINT, CSV, PRINTER, PRINTER_DIALOG, Kein	Festlegung des Protokollformats.
System	Pfadangabe	Angabe eines alternativen Systemverzeichnis
Schluesselpfad	Pfadangabe	Verzeichnisname für die Ablage von benutzerspezifischen Schlüssel
Schluesseldatei	Dateiangabe	Name der Datei für benutzerdefinierte Arbeitsmodi. Standardmäßg wird *Konfig/schluessel.xml* verwendet.
Pruefinfo	Dateiangabe	Optionale Angabe eines Kommunikationssatzes

Schalter	Zulässige Werte	Funktion
Paketgroesse	Numerischer Wert, mit einem Minimum von 1000. Daneben sind folgende Konstanten vordefiniert: - CD (= 650.000.000 Bytes) - Unbegrenzt	Legt die Maximalgröße der zu erstellenden Dateien fest. Voreingestellt ist *Unbegrenzt*
KonfigDialog	Ja / Nein	Hierüber wird festgelegt, ob im Betrieb mit grafischer Oberfläche der Dialog zum Anpassen der Konfiguration aufgerufen werden darf. Voreingestellt ist *Ja*
AbrechnungsKomprimierung	Ja / Nein	Legt fest, ob im Arbeitsmodus ‚Abrechnungsverschluesselung' die automatische Komprimierung aktiviert wird. Voreingestellt ist *Ja*
BearbeitetVerschluesselt	Pfadangabe	Ablage für Dateien, die im Servermodus erfolgreich verschlüsselt wurden
BearbeitetEntschluesselt	Pfadangabe	Ablage für Dateien, die im Servermodus erfolgreich entschlüsselt wurden.
Temp	Pfadangabe oder Konstante ‚TEMP'	Ablage für temporäre Daten, die während der Verarbeitung durch das XKM erzeugt werden. Bei der Angabe ‚TEMP' wird das nutzerspezifische temporäre Verzeichnis verwendet, und um den Pfad ‚KBV/XKM/Work/' erweitert.

Tabelle 5: Konfigurationsschalter

3.4 Arbeiten im Kommandozeilenmodus

Das XKM kann mithilfe von Kommandozeileparametern gesteuert werden, welche die in der Konfigurations-XML (‚config.xml') definierten Schalter überschreiben. Bitte beachten Sie, dass bei einem Aufruf in der Windows-Konsole systembedingt maximal 9 Übergabeparameter möglich sind.

3.4.1 Kommandozeilenparameter

Übergabeparameter	Beschreibung
-c	Name der XKM-Konfigurationsdatei. Default: ‚*Konfig/config.xml*'.
-f	Datei- oder Pfadangabe, welche vom XKM abgearbeitet werden soll. Überschreibt XML-Schalter ‚Quelle'.
-m	Festlegung des Arbeitsmodus. (siehe hierzu 3.2 „Die Arbeitsmodi"). Überschreibt XML-Schalter ‚Arbeitsmodus'
-v	Pfadangabe für das Verschlüsselungsverzeichnis. Überschreibt XML-Schalter ‚Verschluesselt'.
-t	Pfadangabe für das Entschlüsselungsverzeichnis. Überschreibt XML-Schalter ‚Entschluesselt'
-a	Pfadangabe für das Ausschuss-Verzeichnis. Überschreibt XML-Schalter ‚Ausschuss'
-s	Das XKM wird hierdurch im s.g. Servermodus gestartet. Es verarbeitet im 30 Sekunden-Takt Dateien eines Eingangsverzeichnisses. Der Abbruch des Programmlaufs kann über *CTRL-C* bzw. *STRG-C* erfolgen.
-e	Das XKM verarbeitet alle Dateien eines Eingangsverzeichnisses und beendet sich anschließend. (Einzellaufmodus).Schalter nur in Verbindung mit Servermodus setzbar.
-l	Die Eingangsdateien werden nach der Ver- bzw. Entschlüsselung gelöscht.
-p	Name der zu erstellenden Protokolldatei. Überschreibt XML-Schalter ‚Protokolldatei'.
-h	Das XKM gibt einen Hilfetext aus und beendet sich anschließend.
-z	Mittels dieser Option kann dem XKM ein alternatives Arbeitsverzeichnis zugewiesen werden. Überschreibt XML-Schalter ‚System'.
-k	Name der Datei für die Definition zusätzlicher Arbeitsmodi. Überschreibt XML-Schalter ‚Schluesseldatei'. Default: ‚konfig/schluessel.xml'
-u	Verzeichnis für die Ablage von benutzerspezifischen Schlüsseln. Überschreibt XML-Schalter ‚Schluesselpfad'.
-i	Dateiangabe zum Kommunikationssatz. Überschreibt XML-Schalter ‚Pruefinfo'
-g	Festlegen der Maximalgröße eines Pakets. Überschreibt XML-Schalter ‚Paketgroesse'
-o	Möglichkeit zum Konfigurationsdialog ein- bzw. ausschalten. Überschreibt XML-Schalter ‚KonfigDialog'
-r	Festlegen des Protokollformats (PDF, XLS, …). Überschreibt XML-Schalter ‚Protokollformat'
-q	Ein- bzw. Ausschalten der autom. Komprimierung im Modus ‚Abrechnungs_Verschluesselung'
-j	Alternatives Verzeichnis für temporäre Daten. Bei der Angabe ‚TEMP' wird das nutzerspezifische temporäre Verzeichnis verwendet.
-x	Name des Pfads für Bearbeitet-Verschluesselte Dateien im Servermodus. Überschreibt XML-Schalter ‚BearbeitetVerschluesselt'
-y	Name des Pfads für Bearbeitet-Entschluesselte Dateien im Servermodus. Überschreibt XML-Schalter ‚BearbeitetEntschluesselt'

Tabelle 6: Kommandozeilenparameter

3.5 Umgebungsvariablen

Jedes Element der Konfigurationsdatei darf Umgebungsvariablen enthalten.

Diese Umgebungsvariablen müssen der JavaVM jedoch über den Übergabeparameter –D mitgeteilt werden. Nach dem Einlesen der Konfigurationsdatei werden die Umgebungsvariablen durch Ihre Werte ersetzt. Findet das XKM eine Umgebungsvariable nicht, wird das Programm mit einem Konfigurationsfehler abgebrochen.

Mit Hilfe von Umgebungsvariablen kann mehreren Benutzern eine separate Umgebung zur Verfügung gestellt werden, die auf eine einzige Installation zugreift.

Beispiel:

In der Konfigurationsdatei wird der Pfad für die eingehenden und zur Verschlüsselung bestimmten Dateien folgendermaßen festgelegt:

```
<quelle>%DATENEINGANG%/</quelle>
```

Die Umgebungsvariable DATENEINGANG muss dann entweder in einer Batchdatei bzw. einem Shellskript gesetzt werden:

```
set DATENEINGANG=c:\daten\input
```

oder in der aufrufenden Applikation entsprechend gesetzt werden.

Letztlich muss die Umgebungsvariable der JavaVM bekannt gemacht werden. Dies erreicht man durch folgenden Aufruf:

```
java -DDATENEINGANG=%DATENEINGANG% ...
```

3.6 Arbeiten mit grafischer Oberfläche

Alternativ zum Kommandozeilen-Betrieb können Sie viele Funktionen des XKM auch mit der grafischen Oberfläche durchführen. Der Aufruf erfolgt durch *StartGUIKryptomodul.bat* bzw. *StartGUIKryptomodul.sh*.

3.6.1 Der Standard-Dialog

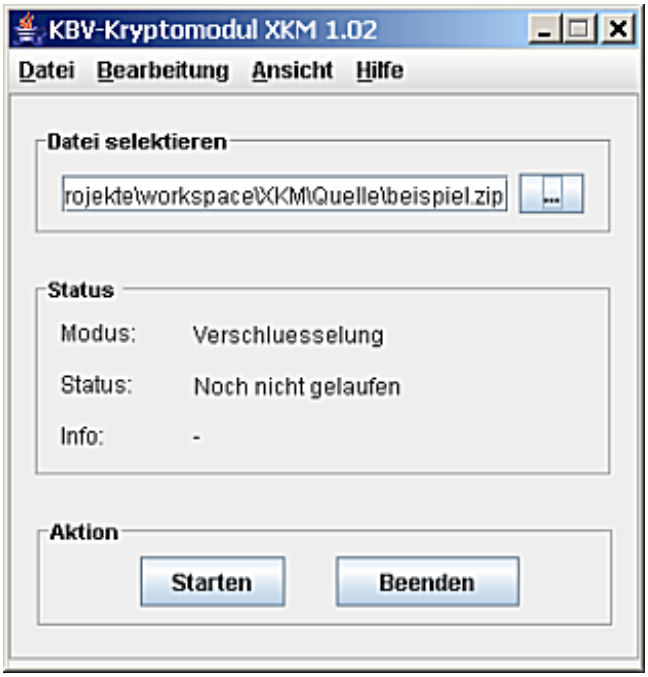

Abbildung 1: Der Standarddialog

3.6.1.1 Datei für die Ver- oder Entschlüsselung selektieren

Bitte drücken Sie den ...-Knopf und wählen Sie die gewünschte Datei aus.

3.6.1.2 Ver- oder Entschlüsselung starten

Bitte drücken Sie den Starten-Knopf.

Nach erfolgter Verarbeitung erscheint eine Meldung mit einem entsprechenden Status.

3.6.1.3 Programm beenden

Bitte drücken Sie den Beenden-Knopf um das XKM zu verlassen.

Diese Option steht Ihnen nur zur Verfügung, wenn das XKM nicht aktiv ist, also keine Ver- oder Entschlüsselung durchführt.

3.6.1.4 Ver- oder Entschlüsselung abbrechen

Eine laufende Verarbeitung können Sie mittels Abbrechen-Knopf vorzeitig abbrechen.

Der Konfigurations-Dialog

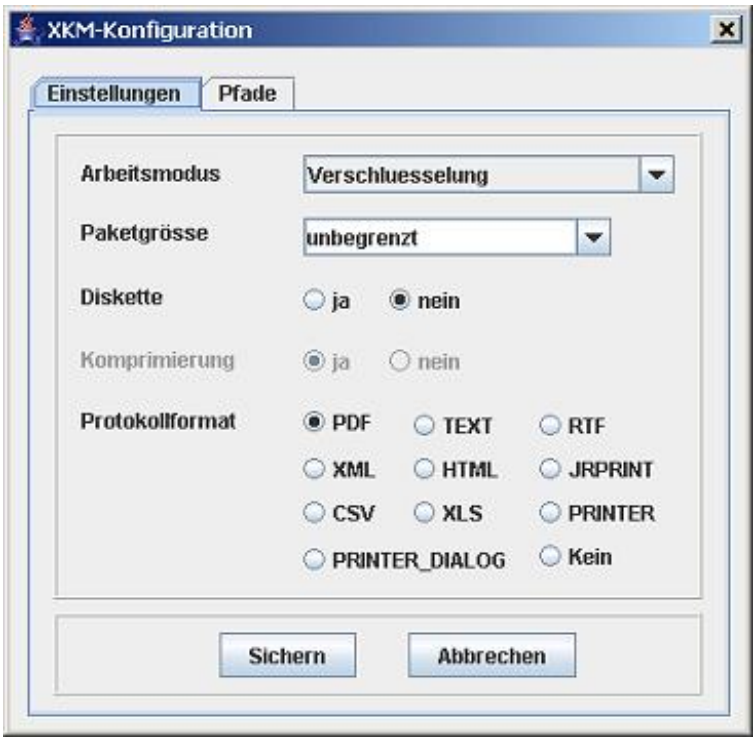

Abbildung 2: XKM Kryptomodul im Konfigurations-Dialog

Den Konfigurations-Dialog erreichen Sie über die Menüabfolge *Bearbeitung->Optionen*. Hier können fast alle Einstellungen, die im Zusammenhang mit den Schaltern der Konfigurationsdatei *Konfig/config.xml* beschrieben wurden, angepasst werden.

Sollte die Möglichkeit zur Optionsverwaltung aus Sicherheitsgründen nicht erwünscht sein, so können Sie in der Konfigurationsdatei über den Schalter *<KonfigDialog>Nein</KonfigDialog>* den Aufruf dieses Dialogs unterbinden.

3.7 Arbeiten mit Wechselmedien (CD, ...)

Das XKM unterstützt den Einsatz von Wechselmedien beispielsweise CD´s. Zudem kann aber auch jedes andere Wechselmedium in den Arbeitsablauf integriert werden. Hierzu müssen die Konfigurationsschalter *Paketgroesse* sowie *Quelle* bzw. *Verschluesselt* geeignet belegt werden. (Siehe hierzu auch das Kapitel *Konfigurationsschalter*)

3.7.1 Notwendige Schritte zur Integration eines Wechselmediums

3.7.1.1 Schalter *Paketgroesse:*

Der Schalter *Paketgroesse* orientiert sich an der Kapazität des jeweiligen Mediums. Hier können Sie im Prinzip beliebige Werte einsetzen. Lediglich 2 Einschränkungen gilt es zu beachten, deren Übertretung vom XKM überwacht wird:

- Die Kapazität eines Mediums muss mindestens 1000 Bytes betragen.
- Die maximale Anzahl an Einzelpaketen liegt bei 999.

Bei gesplitteten Paketen wird zusätzlich zu der Endung *.XKM* noch ein dreistelliges numerisches Suffix angefügt. (*Daten.XKM.001*, *Daten.XKM.002* usw.)

3.7.1.2 Schalter *Quelle* (Verschlüsselung) bzw. *Verschluesselt* (Entschlüsselung)

Der Schalter Quelle/*Verschluesselt* gibt das Verzeichnis an, wo das XKM die verschlüsselten Daten laden und sichern kann.

3.8 Hilfsprogramme des XKMs

Zusätzlich zum XKM-Hauptprogramm gibt es eine Reihe von Hilfsprogrammen, die die Generierung weiterer Schlüsselpaare ermöglichen und die Funktionalität des Hauptprogramms erweitern.

3.8.1 Erzeugen benutzerspezifischer Schlüsselpaare

Dateiname	Funktion
ErzeugeBenutzerschluessel.bat ErzeugeBenutzerschluessel.sh	Generierung eines benutzerspezifischen Schlüsselpaars
ErzeugeGUIBenutzerschluessel.bat ErzeugeGUIBenutzerschluessel.sh	Generierung eines benutzerspezifischen Schlüsselpaars mithilfe eines GUI-Dialogs.

Tabelle 7: Aufruf zum Erzeugen benutzerspezifischer Schlüsselpaare

Die Hilfsprogramme *ErzeugeBenutzerschluessel* und *ErzeugeGUIBenutzerschluessel* sind in ihrer Funktion sehr ähnlich. Im ersten Fall wird auf Konsolenebene mithilfe des Übergabe-Parameters *–b* ein benutzerspezifisches Schlüsselpaar erzeugt, im zweiten Fall wird der Name für das Schlüsselpaar (Benutzerkennung) vor der Generierung im Dialog erfragt.

Zusätzlich zur Schlüsselgenerierung werden in der Datei *System/schluesselintern.xml* zwei weitere Arbeitsmodi (*Verschluesselung* und *Entschluesselung*) zu diesen erzeugten Schlüsseln generiert. Sofern die beiden Arbeitsmodi bereits existieren, erfolgt keine Generierung neuer Schlüssel.

Die Namen der beiden Schlüssel lauten:

Public Key: schluessel/<benutzerkennung>_oeffentlicher_Schluessel.pub

Private Key: system/keys/<benutzerkennung>_privater_Schluessel.NICHT_weitergeben.pfx

3.8.2 Erzeugen allgemeiner Schlüsselpaare

Hinweis: Nicht Bestandteil der Auslieferung, nur für den internen Gebrauch bestimmt.

Dateiname	Funktion
ErzeugeSchluesselpaar.bat ErzeugeSchluesselpaar.sh	Generierung eines Schlüsselpaars mit einem öffentlichen und einem privaten Schlüssel.

Tabelle 8: Aufruf zum Erzeugen allgemeiner Schlüsselpaare

Mit dem Hilfsprogramm *ErzeugeSchluesselpaar* lassen sich beliebig neue Schlüsseldateien generieren. Die Schlüssel werden im Verzeichnis *Schluessel* abgelegt und erhalten als Dateinamen eine Datum/Zeit-Etikette.

Zum Beispiel:

Public Key: Schluessel/2012_06_26_17_31_12_public.pub

Private Key: Schluessel/2012_06_26_17_31_12_private.pfx

4 Starten des XKM

4.1 Starten des XKM von Konsole/Desktop

Zum Starten des XKM gibt es die vordefinierten Batch- bzw. Shellskripte StartGUIKryptomodul und StartKryptomodul. Im Folgenden soll beschrieben werden, wie sie die dort eingestellten Parameter anpassen können.

Das XKM ist eine Java-Applikation und wird in einer Java Laufzeitumgebung ausgeführt.

Hier ein Beispiel für einen Aufruf:

```
java    -Xmx300m
        -Dfile.encoding=8859_1
        -classpath "Bin\bcprov-jdk15on-161.jar;Bin\xkm-1.36.0.jar"
        de.kbv.xkm.Main
        -mVerschluesselung -fquelle\beispieldatei.zip
```

Der Befehl ‚`java`' startet die virtuelle Maschine von Java.

Der Parameter ‚`-Xmx300m`' erlaubt der Java Laufzeitumgebung einen Hauptspeicher von bis zu 300 MB zu reservieren. Diese Option garantiert einen stabilen Programmablauf in der Größenordnung bis ca. 500 MB je Datei.

Der Parameter ‚`-Dfile.encoding=8859_1`' stellt den entsprechenden Zeichensatz ein und ermöglicht hier die Verwendung von deutschen Umlauten.

Der Parameter ‚`-classpath "Bin\bcprov-jdk15on-1.61.jar;Bin\xkm-1.35.0.jar"`' stellt alle Java-Archive zusammen, die für den Programmablauf benötigt werden.

Der Parameter ‚`de.kbv.xkm.Main`' ist der Name der Startklasse.

Die Parameter ‚`-mVerschluesselung`' und ‚`-fquelle\beispieldatei.zip`' sind die eigentlichen Übergabeparameter, die an das XKM übergeben werden.

5 Schlüssel

Jeder vom XKM verwendete Schlüssel existiert als separate PEM-Dateien mit der Endung „.pub“ oder „.pfx“. Zwei zueinander gehörige Teile eines Schlüsselpaars stellen die Endpunkte eines Verschlüsselungsszenarios dar.

Der eine Teil ist der öffentliche Schlüssel (enthält die Bezeichnung „oeffentlich“ als Dateinamensbestandteil), der verteilt und zum verschlüsseln verwendet wird.

Das passende Gegenstück ist der private Schlüssel (enthält die Bezeichnung „privat“ als Dateinamensbestandteil), der zum entschlüsseln verwendet wird und keinesfalls weitergegeben werden darf.

Die beiden Standardschlüsselverzeichnisse sind %XKMROOT%/System/keys/ (hier sollte nichts geändert werden) und %XKMROOT%/Schluessel/, welches zur Ablage nachträglich erhaltener (in der Regel öffentlicher) Schlüssel verwendet werden kann.

5.1 Individuelles Benutzerschlüsselpaar

Das individuelle Benutzerschlüsselpaar kann für beliebige Zwecke jenseits der Standardkommunikationsszenarien verwendet werden. Dabei handelt es sich um ein auf zufälligen Parametern beruhendes, eindeutiges Schlüsselpaar, das auf Dateinamensebene mittels Benutzerkennung „personalisiert“ wird.

Jeder XKM-Instanz steht somit ein individuelles Schlüsselpaar zur Verfügung, dessen öffentlicher Schlüssel (d.h. eine Kopie der entsprechenden Datei) offen weitergereicht werden kann. Die vom XKM des Kommunikationspartners damit verschlüsselten Daten können nur mit dem (ausschließlich in der eigenen Instanz verbleibenden!) privaten Schlüssel wieder entschlüsselt werden. Um an den Kommunikationspartner adressierte Daten verschlüsseln zu können, ist eine Kopie dessen öffentlichen individuellen Schlüssels notwendig.

Beim Start des XKM mit GUI findet eine Überprüfung auf Existenz eines individuellen Schlüsselpaares statt. Ist keines vorhanden, wird per Dialog eine Benutzerkennung abgefragt. Diese Benutzerkennung dient zum Personalisieren der Dateinamen des individuellen öffentlichen und privaten Schlüssels.

Wenn die Konsolenvariante des XKM verwendet wird, muss die Erzeugung eines individuellen Schlüsselpaars selbst erfolgen. Hierzu kann der Datei „ErzeugeBenutzerschluessel.*“ mit Parameter ‚-b’eine Benutzerkennung übergeben werden (alternativ kann die Benutzerkennung auch per Dialog über „ErzeugeGUIBenutzerschluessel.*“ angegeben werden).

Nach Erzeugung eines individuellen Benutzerschlüsselpaars stehen 2 neue Arbeitsmodi zur Verfügung: „Verschluesselung“ und „Entschluesselung“, wobei hauptsächlich letzter relevant ist, um die in einer anderen XKM-Instanz mit dem vorher dorthin übermittelten eigenen öffentlichen Schlüssel verschlüsselte Daten wieder entschlüsseln zu können.

Beispiel: Erzeugung eines individuellen Schlüsselpaars für die Arztnummer bzw. Betriebsstättennummer „123456789“

Variante 1: Verwendung der XKM-Konsolenvariante, Übergabe der Benutzerkennung mit Parameter	• Durch Aufruf von „ErzeugeBenutzerschluessel.*“ mit Parameter „-b123456789“ wird die Schlüsselgenerierung gestartet
Variante 2: Verwendung der XKM-Konsolenvariante, Eingabe der Benutzerkennung über GUI	• Der GUI-Dialog zur Eingabe der Benutzerkennung kann durch Aufruf von „ErzeugeGUIBenutzerschluessel.*“ gestartet werden. Dort ist dann wie in Variante 3 ab dem zweiten Punkt vorzugehen.
Variante 3: Verwendung des XKM mit GUI	• Wenn noch kein individuelles Schlüsselpaar vorliegt, wird automatisch der Eingabedialog für die Benutzerkennung aufgerufen. • Im Eingabefeld wird die Zahlenfolge „123456789“ eingetippt. • Mit dem Schalter „Übernehmen“ wird die Schlüsselgenerierung gestartet.
Privater Schlüssel von Arzt 123456789	• Der private Schlüssel wird mit dem Dateinamen „123456789_privater_Schluessel.NICHT_weitergeben.pfx“ in %XKMROOT%/System/keys/ angelegt. • Der private Schlüssel steht nun unter dem Modusname „Entschluesselung“ zur Verfügung. • Von dem privaten Schlüssel sollte unbedingt eine Sicherungskopie erstellt werden, die an einem sicheren Ort aufbewahrt wird.
Öffentlicher Schlüssel von Arzt 123456789	• Der öffentliche Schlüssel wird mit dem Dateinamen „123456789_oeffentlicher_Schluessel.pub“ in %XKMROOT%/Schluessel/ angelegt. • Der öffentliche Schlüssel steht nun unter dem Modusname „Verschluesselung“ zur Verfügung. • Kopien dieser öffentlichen Schlüsseldatei sind zur Weitergabe an Kommunikationspartner gedacht, die an den Arzt „123456789“ gerichtete Daten (Arztbrief, Befund…) verschlüsseln sollen.

Achtung: Pro Praxis sollte nach Möglichkeit nur ein XKM installiert, und nur ein individuelles Schlüsselpaar angelegt werden, da ansonsten die Verwaltung externer Schlüssel und des Benutzerschlüssels Probleme bereiten kann.

Dieses Problem kann bei einer Netzwerkinstallation (siehe Kapitel 7.3) umgangen werden.

5.2 Privater Schlüssel für zertifizierte Systemanbieter

Um zertifizierten Systemanbietern weiterhin den Zugang zu verschlüsselten Datenlieferungen der KBV zu ermöglichen, erhalten diese einen privaten Schlüssel verschlüsselt per E-Mail. Dieser ist nur zur Verwendung beim Systemanbieter vorgesehen und darf nicht an die Kunden mit ausgeliefert werden.

Einbinden: Datei mit dem privaten Schlüssel („Privat_SWH_Vxy.pfx“) in das Verzeichnis %XKMROOT%/Schluessel/ kopieren. Die folgenden grünmarkierten Einträge in der Datei %XKMROOT%/Konfig/schluessel.xml ergänzen:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Beispieldatei für benutzerdefinierte Arbeitsmodi -->
<arbeitsmodi>
   <arbeitsmodus>
      <typ>entschluesselung</typ>
      <name>Stammdatei_Entschluesselung</name>
      <schluessel>Privat_SWH_V01.pfx</schluessel>
   </arbeitsmodus>
</arbeitsmodi>
```

Beispiel: Entschlüsselung der Kostenträgerstammdatei

Schlüssel einbinden	• Es muss sichergestellt werden, dass der private SWH-Schlüssel wie beschrieben eingebunden ist.
Verschlüsselte Datei besorgen	• Die verschlüsselte Datei (z.B. Xktst741.306.XKM) in das Verzeichnis %XKMROOT%/Verschluesselt/ kopieren.
Variante 1: Verwendung der XKM-Konsolenvariante	• Aufruf von „StartKryptomodul.*“ mit Parameter „-m Stammdatei_Entschluesselung“. Dadurch wird die Entschlüsselung gestartet.
Variante 2: Verwendung des XKM mit GUI	• Aufruf von „StartGUIKryptomodul.*“, um die grafische Oberfläche zu starten • Im Menü „Bearbeitung“ den Menüpunkt „Optionen…“ auswählen (geht nur, wenn in der Konfigurationsdatei %XKMROOT%/Konfig/config.xml der Schalter „konfigdialog“ nicht auf „nein“ gesetzt ist). • In der Listenauswahl den Arbeitsmodus „Stammdatei_Entschluesselung“ auswählen. • Mit dem „Sichern“-Schalter den Optionendialog speichern und beenden. • Im Bereich „Datei selektieren“ mit dem „…“-Schalter den Auswahldialog öffnen. • Die verschlüsselte Stammdatei (z.B. Xktst741.306.XKM) markieren und mit dem „Öffnen“-Schalter auswählen. • Im Bereich „Aktion“ mit dem „Starten“-Schalter die Entschlüsselung starten.
Entschlüsselte Datei	• Die entschlüsselte Datei (Xktst741.306) wird in dem Verzeichnis %XKMROOT%/Entschluesselt erstellt.
Statusbericht	• Ein Statusbericht zum Entschlüsselungsvorgang wird im Listenverzeichnis in dem in der Konfigurationsdatei eingestellten Ausgabeformat erzeugt.

5.3 Schlüssel für die Standardkommunikationsszenarien den Praxen

Für die Datenverschlüsselung im Rahmen der Abrechnung, bei DMP und Koloskopie, so wie für die Auftrags- und Befundverschlüsselung und -entschlüsselung gibt es vorgefertigte Arbeitsmodi. D.h. die Zuordnung eines Modusnamens, -typs und weiterer Attribute zu einem Schlüsseldateinamen existiert in der Datei %XKMROOT%/System/schluesselintern.xml bereits. Im XPM-Paket stehen diese Standardmodi zunächst noch nicht zur Verfügung, da (anders als beim Prüfassistent und beim Kryptomat) noch keine Schlüssel enthalten sind.

Wichtig: Vor Auslieferung des XKM-Pakets muss es mit den für den Endanwender relevanten Schlüsseln bestückt werden.

Beispiel: Vorbereitung des XKM-Pakets für Abrechnungsverschlüsselung und Auftrags- und Befundverschlüsselung und –entschlüsselung.

Schlüssel besorgen	• Laden Sie von unserer Internetseite (http://www.kbv.de/html/updates.php) die notwendigen Schlüssel in der aktuell gültigen Version herunter: - Abrechnungsverschlüsselung: Oeffentlich_KV_V08.pub
Einbinden	• Kopieren Sie die Schlüssel in das Verzeichnis %XKMROOT%/System/keys/ • Stellen Sie sicher, dass die Dateinamen der heruntergeladenen Schlüssel mit den entsprechenden Einträgen der Elemente „schluessel“ in der Datei „schluesselintern.xml (in %XKMROOT%/System/) übereinstimmen und passen Sie die Einträge ggf. an. • Nun stehen die Modi unter den folgenden Modusnamen zur Verfügung: - Abrechnungs_Verschluesselung

6 Schlüsselverwaltung / Arbeitsmodi

Die Schlüsselverwaltung und –zuordnung zu den entsprechenden Modi erfolgt über 2 XML-Dateien: schluesselintern.xml (Standardverzeichnis: %XKMROOT%/System/) und schluessel.xml (Standardverzeichnis: %XKMROOT%/Konfig/). Die Datei „schluesselintern.xml“ sollte im Normalfall nicht editiert werden, und muss sich immer in dem Systemverzeichnis befinden, dessen Pfad und Dateiname konfigurierbar ist. Erweiterungen können über die Datei „schluessel.xml“ definiert werden, deren Pfad und Dateiname konfigurierbar ist.

6.1 Einbinden von Schlüsseln

Die Standardarbeitsmodi sind bereits in der Datei „schlüsselintern.xml“ definiert. Weitere Schlüssel können in der Datei „schluessel.xml“ (Standardverzeichnis %XKMROOT%/Konfig/) eingebunden werden. Dort sind bereits 2 auskommentierte Musterblöcke (Element „Arbeitsmodus“ mit Kindelementen) enthalten.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Beispieldatei für benutzerdefinierte Arbeitsmodi -->

<arbeitsmodi>

  <!--  Beginn der ersten Auskommentierung -
  <arbeitsmodus>
    <typ>verschluesselung</typ>
    <name>Spezial1_Verschluesselung</name>
    <schluessel>Public_Key_allgemein_V01.pub</schluessel>
  </arbeitsmodus>
  -  Ende der ersten Auskommentierung -->

  <!--  Beginn der zweiten Auskommentierung -
  <arbeitsmodus>
    <typ>entschluesselung</typ>
    <name>Spezial1_Entschluesselung</name>
    <schluessel>Private_Key_allgemein_V01.pfx</schluessel>
  </arbeitsmodus>
  -   Ende der zweiten Auskommentierung -->

</arbeitsmodi>
```

Im Element „typ“ wird die Verschlüsselungsart festgelegt, es gibt die beiden Typen „verschluesselung“ und „entschluesselung“. Ein einzubindender öffentlicher Schlüssel sollte immer einem Arbeitsmodus vom Typ „verschluesselung“, ein privater Schlüssel immer vom Typ „entschluesselung“ sein.

Im Element „name“ wird ein frei wählbarer Modusname festgelegt. Unter diesem Namen (Übergabe mit Parameter ‚-m’, oder Auswahl auf dem GUI-Optionendialog) erfolgt programmintern die Schlüsselauswahl. Dieser Name muss eindeutig sein (d.h. darf in den Dateien schluesselintern.xml und schluessel.xml insgesamt nur einmal vorkommen), ansonsten meldet das XKM beim nächsten Startversuch einen Konfigurationsfehler und kann nicht gestartet werden.

Im Element „schluessel“ wird der Dateiname des einzubindenden Schlüssels angegeben. Der einzubindende Schlüssel muss physikalisch unter dem angegebenen Dateinamen „Schluessel“ (Standardpfad %XKMROOT%/Schluessel/) abgelegt sein, erst dadurch steht der Modus unter der im Element „name“ festgelegten Bezeichnung auch tatsächlich zur Verfügung.

6.2 Testen von Ver- und Entschlüsselung

Als Hilfestellung sowie zur Verwendung im Rahmen von internen Qualitätssicherungen der Softwarehäuser wurde die Datei „schluesselintern.xml“ um die Arbeitsmodi TEST_Verschluesselung und TEST_Entschluesselung erweitert. Zugehörig zu den Arbeitsmodi wurden die Schlüssel Privat_Testschluessel_V01.pfx und Oeffentlich_Testschluessel_V01.pub erzeugt.

Hinweis: Die Arbeitsmodi sowie das Schlüsselpaar sind nur zu TESTZWECKEN zu verwenden.

Im Rahmen der Anwendung erfolgt durch das XKM ein Hinweis auf Konsole bzw. im Protokoll, dass die erstellte Datei nur für Testzwecke zu verwenden ist.

Ver-/Entschlüsselungsprotokoll Kryptomodul (XKM)

Dateiname der Quelle:	D:\ITA\XKM1.17\XKM\Verschluesselt\Z01129990000_20.08.2010_12.00.con.XKM
Dateiname der Zieldatei:	D:\ITA\XKM1.17\XKM\Entschluesselt\Z01129990000_20.08.2010_12.00.con
Entschlüsselungsdatum:	19.01.2012
Entschlüsselungsuhrzeit:	11:49:48
Anzahl Pakete:	1
Arbeitsmodus:	TEST_Entschluesselung
Verwendeter Schlüssel:	System/keys/Privat_Testschluessel_V01.key
Verwendete XKM-Version:	1.17

*** Errorlevel - Endstatus: 0 - ok ***

Die erstellte Datei darf nur zu Testzwecken verwendet werden (Testschlüssel) !!!

Abbildung 3: Protokoll Test-Ver-/Entschlüsselung

```
c:\JavaProjekte\workspace\xkm\Doku>"Anwendungshinweise XKM.doc"

c:\JavaProjekte\workspace\xkm>StartKryptomodul.bat -mTEST_Verschluesselung
Starte XKM-Konsole
Bearbeite Datei: xyz.bak
Verschluessele ...
Testschluessel-Warnung !
Die Datei wird mit einem Testschluessel erstellt
und darf ausschliesslich zum Testen und NICHT fuer
echte Datenlieferungen verwendet werden!
Schreibe Header ...
Kryptomodul mit Returncode 0 beendet
```

Abbildung 4: Konsole Test-Ver-/Entschlüsselung

7 Konfigurations- und Handhabungsempfehlungen

7.1 Backup der Schlüssel und der Schlüsselzuordnung

Der Prüfassistent verwendet nur einen Schlüssel (für die Abrechnungsverschlüsselung), beim Kryptomat (s. Kapitel 2) sorgt die Installationsroutine dafür, dass das „Schluessel“-Verzeichnis und die Zuordnungsdatei „schluessel.xml“ bei einem Programmupdate nicht überschrieben werden. Dies ist wichtig, um den individuellen privaten und andere eingebundene Schlüssel nicht zu verlieren. Beim XKM-Paket muss selbst für das entsprechende Backup gesorgt werden.

Grundsätzlich sollte bei allen Programmvarianten, wenn außer den Standardschlüsseln zusätzliche Schlüssel hinzugefügt wurden, regelmäßig Sicherungskopien des „System“-Verzeichnisses, des „Schluessel“-Verzeichnisses und der Zuordnungsdatei „schluessel.xml“ (im Verzeichnis „Konfig“) erfolgen, da bei einem Verlust der Daten alle zusätzlichen Schlüssel neu besorgt, und alle zusätzlichen Modi wieder neu definiert werden müssen. Ein Verlust des individuellen privaten Schlüssels führt dazu, dass individuell verschlüsselte Dateien nicht wiederhergestellt werden können.

7.2 GUI-Optionen für Endanwender sperren

Um eine Fehlkonfiguration des XKM durch den Endanwender zu verhindern, kann der Schalter „konfigdialog“ in der Konfigurationsdatei (Standardverzeichnis %XKMROOT%/Konfig/config.xml) auf „nein“ gesetzt sein:

```
<konfigdialog>nein</konfigdialog>
```

Dadurch kann der Menüpunkt „Optionen“ im Menü „Bearbeitung“ nicht ausgewählt werden.

Alternativ müssen dem Anwender dann aber entsprechende Programmstartmöglichkeiten zur Verfügung stehen, mit denen er das XKM in den für ihn relevanten Modi aufrufen kann.

7.3 XKM-Installation auf einem Netzlaufwerk

Wenn das XKM auf einem Netzlaufwerk installiert werden soll, auf das von anderen Rechnern über Netzwerk zugegriffen wird, so muss das „System“-Verzeichnis lokal auf sämtliche Rechner kopiert werden von denen aus das XKM verwendet werden soll. Auf diese Verzeichnisse greift die lokal laufende Java-VM permanent zu.

Da das XKM nicht multi-user fähig ist, würde sonst bei nur einem zentralen Systemverzeichnis bei gleichzeitiger Benutzung von zwei oder mehr Arbeitsplätzen aus entweder ein Fehler auftreten und das Kryptomodul würde abbrechen oder es könnten falsche Daten, welche nicht von dem Anwender selber stammen, verschlüsselt werden.

Hierbei ist zu beachten, dass das „System“-Verzeichnis erst nach einem erstmaligem Aufruf der GUI-Variante bzw. nach manuellem Aufruf von „ErzeugeBenutzerschluessel.*“ kopiert werden sollte, damit alle Installationen denselben individuellen privaten Schlüssel verwenden.

7.4 Starten des XKM aus einem Java-Programm heraus

Das XKM ist eine Java-Applikation und kann von einem anderen Java-Programm aufgerufen werden. Für eine leichtere Anbindung wurde die Klasse de.kbv.xkm.extern.XKMEinstieg implementiert. Diese Klasse ist im Java-Archiv XKM.jar enthalten, und liegt als Source im Dokumentationsverzeichnis vor. Das Setzen und Auslesen der Programmschalter erfolgt hier über entsprechende get/set-Methoden. Anbei ein beispielhafter Aufruf des XKMs über die Klasse XKMEinstieg:

```
package XKMEinstiegBeispiel;
import de.kbv.xkm.extern.XKMEinstieg;
import de.kbv.xkm.XKMException;

public class XKMEinstiegBeispiel {

    public static void main (String args[]) {

        XKMEinstieg xkm = new XKMEinstieg ();
        int nRetCode = 0;

        System.out.println("Kryptomodul ueber XKMEinstieg gestartet");

        try {
            xkm.setQuelle("quelle/beispiel_daten.zip");
            xkm.setProtokolldatei("listen/MeinProtokoll");
            xkm.setProtokollformat("XLS");
            xkm.setArbeitsmodus("Verschluesselung");
            nRetCode = xkm.start();
            System.out.println("XKM mit Returncode " + nRetCode + " beendet");
        }
        catch (XKMException ex){
            nRetCode = ex.getCode();
            System.err.println("Fehlercode: " + nRetCode + ". " + ex.getMessage());
        }
    }
}
```

8 Das Verschlüsselungsverfahren des XKM

Das XKM verwendet ein hybrides Verschlüsselungsverfahren, bei dem die Daten symmetrisch, und der zugehörige symmetrische Schlüssel asymmetrisch verschlüsselt werden. Hierdurch werden die Vorteile der beiden Verfahren auf geeignete Weise kombiniert.

Einige Merkmale des hybriden Verschlüsselungsverfahrens:

a) Performance: Die Nutzdaten werden symmetrisch verschlüsselt, was etwa um den Faktor 1000 schneller als eine asymmetrische Verschlüsselung durchgeführt werden kann. Die asymmetrische Verschlüsselung des symmetrischen Schlüssel („Schlüsselverschlüsselung“) fällt dagegen nicht ins Gewicht.

b) Sicherheit: Das symmetrische Verfahren wird anhand von AES bei einer Schlüssellänge von 128 Bit durchgeführt. Die Größe des asymmetrischen Schlüssels liegt bei 2048 Bit und wird durch das Verfahren RSA realisiert. Die verwendeten Schlüssellängen orientieren sich am aktuellen Stand der Technik.

c) Schlüssellogistik: Die Zahl der nötigen Schlüssel wächst linear mit der Zahl der Teilnehmer. Bei rein symmetrischen Systemen ist der Zuwachs dagegen quadratisch.

d) Eine geheime Übermittlung des Schlüssels - wie bei symmetrischen Verfahren erforderlich – entfällt. Die Kombination aus Private- und Public-Key ermöglicht spontane Kommunikation.

9 Aufbau des XKM-Header

Es folgt eine Strukturbeschreibung des 256-Byte grossen XKM-Header, der jedem verschlüsselten Paket vorangestellt wird:

Pos	Feld	Länge	Offset	Bedeutung
1	Arztnummer bzw. Betriebstättennummer (BSNR)	40	0	Text ‚Arztnummer\BSNR:' + Arztnummer bzw. BSNR (Optional)
2	Zeilenvorschub	2	40	CarriageReturn+LineFeed
3	Erstellungsdatum	40	42	Text ‚Erstellungsdatum:' + Erstellungsdatum im Format dd.mm.yyyy hh:mm:ss
4	Zeilenvorschub	2	82	CarriageReturn+LineFeed
5	Paketangabe	40	84	Text ‚Paket x/y', mit x = aktuelle Paketnummer, y = Pakete insgesamt
6	Zeilenvorschub	4	124	2*CarriageReturn+LineFeed
7	XKM-Signatur	5	128	‚*XKM*' als Erkennungskonstante für eine mit XKM verschlüsselte Datei
8	Zeilenvorschub	2	133	CarriageReturn+LineFeed
9	Versionsnummer	6	135	Zum Beispiel ‚1.08 '
10	Zeilenvorschub	2	141	CarriageReturn+LineFeed
11	Paket-Aktuell	5	143	Aktuelle Paketnummer
12	Zeilenvorschub	2	148	CarriageReturn+LineFeed
13	Paket-Gesamt	5	150	Anzahl Pakete insgesamt
14	Zeilenvorschub	2	155	CarriageReturn+LineFeed
15	Arbeitsmodus	2	157	Index des Arbeitsmodus
16	Zeilenvorschub	2	159	CarriageReturn+LineFeed
17	Komprimierung	1	161	1=Komprimierung, sonst 0 (ab XKM 1.06)
18	Zeilenvorschub	2	162	CarriageReturn+LineFeed
19	Pruefinfo	1	164	1=Kommunikationssatz enthalten, sonst 0 (ab XKM 1.08)
20	Zeilenvorschub	2	165	CarriageReturn+LineFeed
21	Testschlüssel-Flag	4	167	Konstante ‚TEST' bei Testschlüsselung-Verwendung (ab XKM 1.08)
22	Zeilenvorschub	2	171	CarriageReturn+LineFeed
23	Reserviert	83	173	Raum für zukünftige Erweiterungen

Tabelle 9: Strukturbeschreibung des XKM Headers

Hinweise:

- Alle Inhalte werden linksbündig in die Felder geschrieben
- Bei Bedarf wird mit Leerzeichen aufgefüllt