Gesellschaft für Telematikanwendungen der Gesundheitskarte mbH

Einführung der Gesundheitskarte

Spezifikation

KOM-LE-Clientmodul

Version:	1.3.0
Revision	\main\rel_online\rel_ors1\3
Stand:	24.07.2015
Status:	freigegeben
Klassifizierung:	öffentlich
Referenzierung:	[gemSpec_CM_KOMLE]

Dokumentinformationen

Änderungen zur Vorversion

Begriff Betreiber durch Anbieter ersetzt

Dokumentenhistorie

Version	Stand	Kap./ Seite	Grund der Änderung, besondere Hinweise	Bearbeitung
0.5.0	19.11.13		zur Abstimmung freigegeben	gematik
1.0.0	27.01.14		Einarbeitung Kommentare	
1.1.0	28.02.14	4.1.2	XP-Verweis entfernt	P74
1.2.0	25.07.14	3.1 4.1.2/4.1.4	Zeitsynchronisation Konnektor ergänzt Formulierungsanpassungen	P74
			Begriff Betreiber durch Anbieter ersetzt	
1.3.0	24.07.15		freigegeben	gematik

© gematik - öffentlich

Inhaltsverzeichnis

Dokumentinformationen **2**

Inhaltsverzeichnis **3**

1 Einordnung des Dokumentes **5**

1.1 Zielsetzung **5**

1.2 Zielgruppe **5**

1.3 Geltungsbereich **5**

1.4 Arbeitsgrundlagen **5**

1.5 Abgrenzung des Dokuments **6**

1.6 Methodik **7**

1.6.1 Anforderungen 7
1.6.2 Diagramme 7
1.6.3 Nomenklatur 8

2 Systemüberblick **9**

3 Produktfunktionen **11**

3.1 Allgemeine Anforderungen **11**

3.2 Senden von Nachrichten **12**

3.2.1 Übersicht 12
3.2.2 CONNECT-Zustand 14
3.2.2.1 Initialisierung *14*
3.2.2.2 Verbindungsaufbau mit MTA *15*
3.2.3 PROXY-Zustand 19
3.2.4 PROCESS-Zustand 19
3.2.4.1 Empfang und Weiterleitung einer Nachricht *20*
3.2.4.1.1 Bearbeitung einer ungeschützten Nachricht 20
3.2.4.1.2 Bearbeitung einer geschützten KOM-LE-Nachricht 28
3.2.5 Beispiele 29

3.3 Empfangen von Nachrichten **32**

3.3.1 Übersicht 32
3.3.2 CONNECT-Zustand 34
3.3.2.1 Initialisierung *34*
3.3.2.2 Verbindungsaufbau mit dem POP3-Server *35*
3.3.3 PROXY-Zustand 39
3.3.4 PROCESS-Zustand 40
3.3.4.1 Empfang und Weiterleitung einer Nachricht *40*
3.3.4.2 Aufbereitung einer Nachricht *40*

3.3.4.2.1 Entschlüsselung 41
3.3.4.2.2 Integritätsprüfung 44
3.3.5 Beispiele 48
3.4 Übermittlung von Kontaktdaten 49
3.5 Kryptographischen Schnittstellen des Konnektors 50
3.5.1 Erstellung der digitalen Signatur einer Nachricht mit einer SM-B 50
3.5.2 Prüfung der digitalen Signatur einer Nachricht 53
3.5.3 Verschlüsselung einer Nachricht 53
3.5.4 Entschlüsselung einer Nachricht mit einer SM-B bzw. einem HBA 53
4 Nichtfunktionale Anforderungen 57
4.1 Transportsicherung 57
4.1.1 Allgemeine Festlegungen 57
4.1.2 Transportsicherung zwischen Clientsystem und Clientmodul 58
4.1.3 Transportsicherung zwischen Clientmodul und Konnektor 59
4.1.4 Transportsicherung zwischen Clientmodul und Fachdienst 60
4.2 Nutzung von Webservice-Schnittstellen des Konnektors 60
4.3 Protokollierung/Logging 61
4.3.1 Ablaufprotokoll 62
4.3.2 Performance 63
4.3.3 Fehler 64
4.4 Konfiguration 65
4.5 Update-Mechanismen 66
4.6 Produktleistungen 66
4.6.1 Performance 66
4.6.2 Skalierbarkeit 66
Anhang A 67
A1 – Abkürzungen 67
A2 – Glossar 67
A3 – Abbildungsverzeichnis 68
A4 – Tabellenverzeichnis 68
A5 - Referenzierte Dokumente 69
A5.1 – Dokumente der gematik 69
A5.2 – Weitere Dokumente 69

1 Einordnung des Dokumentes

1.1 Zielsetzung

Das vorliegende Dokument spezifiziert die Anforderungen an den Produkttyp KOM-LE-Clientmodul. Das Clientmodul ist verantwortlich für das Signieren und Verschlüsseln von KOM-LE-Nachrichten beim Versenden sowie für die Entschlüsselung und Signaturprüfung beim Abholen von KOM-LE-Nachrichten.

Aus den Kommunikationsbeziehungen mit Clientsystem, Konnektor, Verzeichnisdienst und KOM-LE-Fachdienst resultieren vom Clientmodul anzubietende Schnittstellen, die in diesem Dokument normativ beschrieben werden. Vom Clientmodul genutzte Schnittstellen liegen zumeist in anderen Verantwortungsbereichen (Konnektor, Verzeichnisdienst). Diese werden in den entsprechenden Produkttypspezifikationen definiert.

1.2 Zielgruppe

Dieses Dokument richtet sich an

- Entwickler des KOM-LE-Clientmoduls,
- Primärsystemhersteller und
- Verantwortliche für Zulassung und Test.

1.3 Geltungsbereich

Dieses Dokument enthält normative Festlegungen zur Telematikinfrastruktur des deutschen Gesundheitswesens. Der Gültigkeitszeitraum der vorliegenden Version und deren Anwendung in Zulassungsverfahren werden durch die gematik GmbH in gesonderten Dokumenten (z.B. Dokumentenlandkarte, Produkttypsteckbrief, Leistungsbeschreibung) festgelegt und bekannt gegeben.

1.4 Arbeitsgrundlagen

Grundlagen für die Ausführungen dieses Dokumentes sind

- Lastenheft Adressierte Kommunikation Leistungserbringer
- Systemspezifisches Konzept KOM-LE [gemSysL_KOMLE]
- KOM-LE S/MIME-Profil [gemSMIME_KOMLE]
- Gesamtarchitektur der TI [gemÜK_Arch_TI]

- Konzept Architektur der TI-Plattform [gemKPT_Arch_TIP]
- Spezifikation PKI [gemSpec_PKI]
- Verwendung kryptographischer Algorithmen in der Telematikinfrastruktur [gemSpec_Krypt]
- Spezifikation Konnektor [gemSpec_Kon]

1.5 Abgrenzung des Dokuments

Spezifiziert werden in dem Dokument die vom Produkttyp bereitgestellten (angebotenen) Schnittstellen. Benutzte Schnittstellen werden hingegen in der Spezifikation desjenigen Produkttypen beschrieben, der diese Schnittstelle bereitstellt. Auf die entsprechenden Dokumente wird referenziert.

Die Systemlösung der Fachanwendung KOM-LE ist im systemspezifischen Konzept [gemSysL_KOMLE] beschrieben. Dieses Konzept setzt die fachlichen Anforderungen des Lastenheftes auf Systemebene um, zerlegt die Fachanwendung KOM-LE in die zugehörigen Produkttypen, darunter das KOM-LE-Clientmodul und der KOM-LE-Fachdienst. Ferner definiert es die Schnittstellen zwischen den einzelnen Produkttypen. Für das Verständnis dieser Spezifikation wird die Kenntnis von [gemSysL_KOMLE] vorausgesetzt.

Die Anforderungen am Fachdienst werden separat in der Spezifikationen Fachdienst KOM-LE [gemSpec_FD_KOMLE] beschrieben.

Die Anforderungen an das Format der KOM-LE-Nachrichten, die zwischen dem Clientmodul und dem Fachdienst übermittelt werden, werden separat im KOM-LE-S/MIME-Profil [gemSMIME_KOMLE] beschrieben.

Abbildung 1 zeigt schematisch die Einbettung des vorliegenden Dokuments in die Dokumentenlandschaft der Lastenheft- und Pflichtenheftphase in Form einer Dokumentenhierarchie.

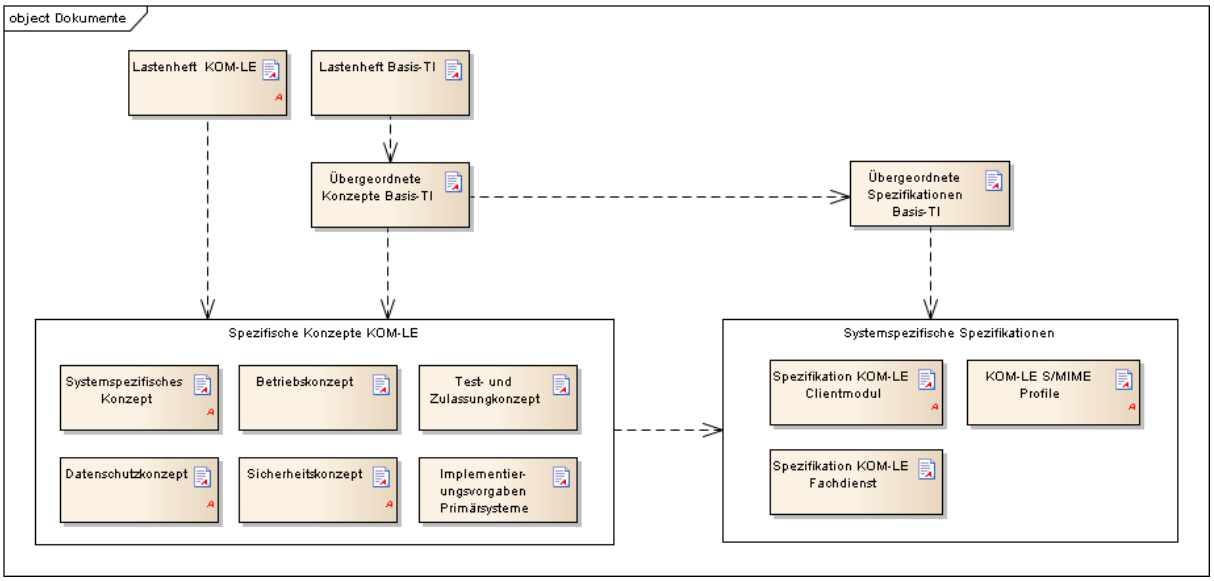

Abbildung 1: Abb_Dok_Hierarchie Dokumentenhierarchie KOM-LE

1.6 Methodik

1.6.1 Anforderungen

Anforderungen als Ausdruck normativer Festlegungen werden durch eine eindeutige ID sowie die dem RFC 2119 [RFC2119] entsprechenden, in Großbuchstaben geschriebenen deutschen Schlüsselworte MUSS, DARF NICHT, SOLL, SOLL NICHT, KANN gekennzeichnet.

Sie werden im Dokument wie folgt dargestellt:

⌦ **TIP1-A_0000 <Titel der Afo>**

Text / Beschreibung⌫

Dabei umfasst die Anforderung sämtliche innerhalb der Textmarken angeführten Inhalte.

1.6.2 Diagramme

Die Darstellung der Spezifikationen von Komponenten erfolgt auf der Grundlage einer durchgängigen Use-Case-Modellierung als

- technische Use Cases (eingebundene Graphik sowie tabellarische Darstellung mit Vor- und Nachbedingungen gemäß Modellierungsleitfaden),
- Sequenz- und Aktivitätendiagramme sowie
- Klassendiagramme
- XML-Strukturen und Schnittstellenbeschreibungen.

1.6.3 Nomenklatur

Sofern im Text dieser Spezifikation auf die Ausgangsanforderungen verwiesen wird, erfolgt dies in eckigen Klammern, z.B. [KOMLE-A_2015]. Wird auf Eingangsanforderungen verwiesen, erfolgt dies in runden Klammern, z.B. (KOMLE-A_202).

© gematik - öffentlich

2 Systemüberblick

Das Clientmodul bietet die Funktionalität, die für Anwendungsfälle KOM-LE_AF_1 „Nachricht senden“ und KOM-LE_AF_2 „Nachricht empfangen“ (siehe [gemSysL_KOMLE]) relevant ist. Die Aufgabe des Clientmoduls ist das Aufbringen und Aufheben des Schutzes der Integrität und Vertraulichkeit der zwischen den KOM-LE-Teilnehmern ausgetauschten E-Mail-Nachrichten. Dabei kommuniziert das Clientmodul mit dem Clientsystem, dem KOM-LE-Fachdienst und nutzt mehrere Dienste der TI-Plattform. Abbildung 2 stellt die grundlegenden Elemente der KOM-LE-Architektur dar.

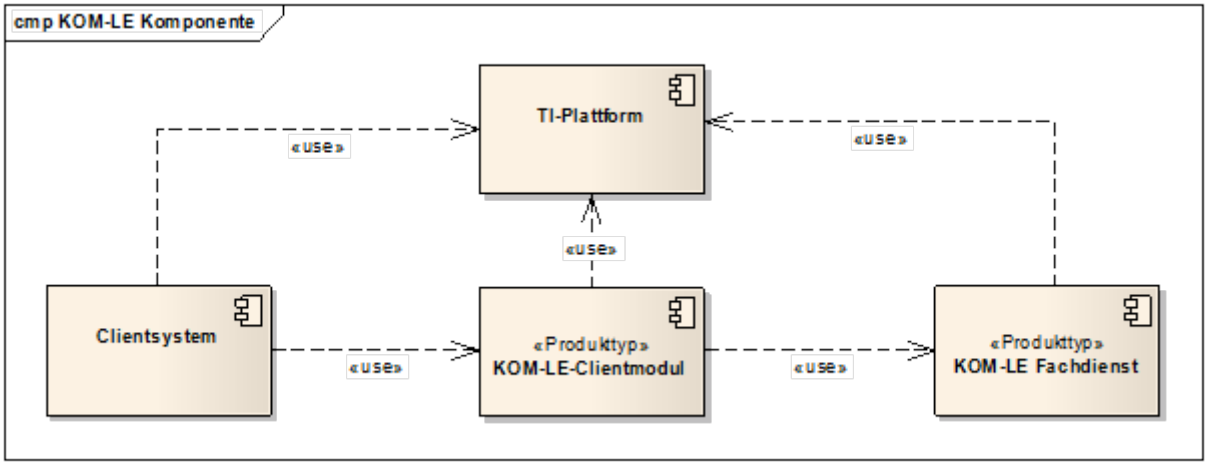

Abbildung 2: Abb_KOMLE_Komp KOM-LE-Komponenten

Die im Clientmodul bearbeitende E-Mail-Nachrichten werden beim Senden entsprechend dem KOM-LE-S/MIME-Profil [gemSMIME_KOMLE] digital signiert und verschlüsselt und beim Empfangen entschlüsselt und deren Signatur geprüft. Das KOM-LE-S/MIME-Profil konkretisiert die S/MIME-Spezifikation und stellt sicher, dass die Interoperabilität zwischen den verschiedenen KOM-LE-Komponenten sowie der Schutz von Integrität und Vertraulichkeit für alle personbezogenen medizinischen Daten gewährleistet werden.

Jede dem KOM-LE-S/MIME-Profil entsprechende Nachricht hat die in Abbildung 3 dargestellte Struktur. Die äußere Nachricht ist eine entsprechend dem S/MIME-Standard signierte und verschlüsselte E-Mail-Nachricht. Die innere Nachricht ist eine im Clientsystem erzeugte E-Mail-Nachricht, die Nutzdaten enthält und als `message/rfc822` Anhang in die äußere Nachricht verpackt ist.

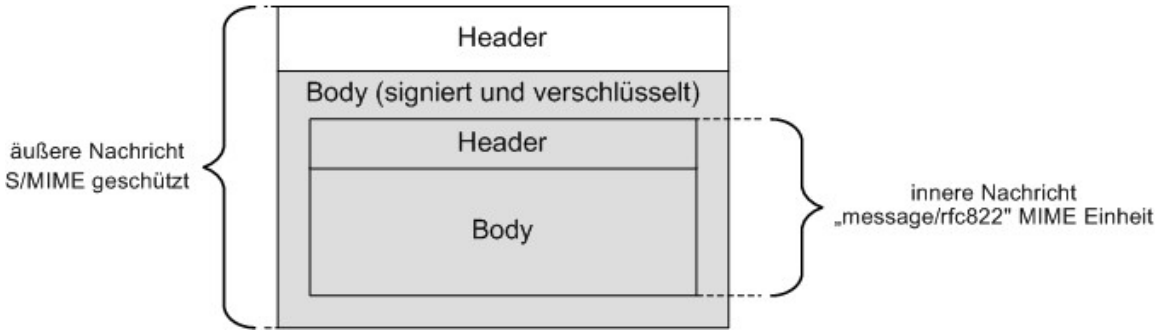

Abbildung 3: Abb_Struk_KOMLE_Msg Struktur einer KOM-LE-Nachricht

3 Produktfunktionen

3.1 Allgemeine Anforderungen

⌦ **KOM-LE-A_2003 Unterstützung von E-Mail-Clients**

Das KOM-LE-Clientmodul MUSS das Senden und Empfangen von Nachrichten mit marktüblichen SMTP/POP3 Desktop-E-Mail-Clients unterstützen. ⌫

⌦ **KOM-LE-A_2004 Größe einer KOM-LE-Nachricht**

Das KOM-LE-Clientmodul MUSS Nachrichten mit einer Nettogröße von bis zu 25 MB bearbeiten können. Dabei ist zu beachten, dass sich durch die base64-Kodierung der Nachricht die zu verarbeitende Bruttogröße um den Faktor 1,37 erhöht. ⌫

⌦ **KOM-LE-A_2005 Keine persistente Speicherung von Nachrichten**

Das KOM-LE-Clientmodul DARF NICHT die Inhalte von Nachrichten länger als es für die Aufbereitung und Übermittlung nötig ist, speichern. ⌫

⌦ **KOM-LE-A_2230 Synchronisation mit der Systemzeit des Konnektors**

Das KOM-LE-Clientmodul MUSS sich unter Verwendung der Operation sync_Time mit der Systemzeit des Konnektors synchronisieren. ⌫

Diese Spezifikation erläutert nicht alle Schritte und Einzelheiten der SMTP- und POP3-Kommunikation zwischen dem Clientsystem, dem KOM-LE-Clientmodul und dem KOM-LE-Fachdienst. Es setzt voraus, dass das Format einer E-Mail, MIME, SMTP und POP3 dem Leser bekannt sind.

⌦ **KOM-LE-A_2006 Einzuhaltende Standards beim Senden und Empfangen**

Das KOM-LE-Clientmodul MUSS sich beim Senden und Empfangen von Nachrichten konform zu folgenden Standards verhalten:

- IETF Draft: The LOGIN SASL Mechanism, K. Murchison, M. Crispin, August 2003,
- RFC 1939: Post Office Protocol – Version 3 [RFC1939],
- RFC 2045: Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies [RFC2045],
- RFC2046: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types [RFC2046],
- RFC 2449: POP3 Extension Mechanism [RFC2449],
- RFC 3463: Enhanced Mail System Status Codes [RFC3463],

- RFC 4616: The PLAIN Simple Authentication and Security Layer (SASL) Mechanism, K. Zeilenga, August 2006 [RFC4616],
- RFC 4954: SMTP Service Extension for Authentication [RFC5321],
- RFC 5321: Simple Mail Transfer Protocol [RFC5248],
- RFC 5322: Internet Message Format [RFC5322],
- RFC 5750: Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Certificate Handling, B. Ramsdell, S. Turner, Januar 2010 [RFC5750] und
- RFC 5751: Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification, B. Ramsdell, S. Turner, Januar 2010 [RFC5751]. ⌫

3.2 Senden von Nachrichten

In diesem Kapitel werden Anforderungen an das Clientmodul formuliert, die für den Anwendungsfall „KOM-LE_AF_1 Nachricht senden“ [gemSysL_KOMLE] spezifisch sind.

3.2.1 Übersicht

Beim Senden von KOM-LE-Nachrichten sorgt das Clientmodul dafür, dass die gesendeten E-Mail-Nachrichten digital signiert und verschlüsselt dem Message Transfer Agent des KOM-LE-Fachdienstes (weiter im Text als MTA bezeichnet), bei dem der Sender registriert ist, übermittelt werden. Abbildung 4 stellt die Interaktionen zwischen den am Senden von KOM-LE-Nachrichten beteiligten Komponenten dar. Aus der Sicht des Clientsystems agiert das Clientmodul als ein MTA und aus der Sicht des MTAs des Fachdienstes agiert das Clientmodul als MUA. Für Funktionen wie Datentransport, kryptographische Operationen und Kommunikation mit dem Verzeichnisdienst verwendet das Clientmodul entsprechende Dienste der TI-Plattform.

Abbildung 4: Abb_Send_Msg Senden von Nachrichten

Beim Senden von Nachrichten findet die Kommunikation zwischen dem Clientsystem, dem Clientmodul und dem MTA über SMTP statt. Das Clientmodul fungiert als SMTP Proxy, der das Clientsystem mit dem MTA verbindet, die Integrität und Vertraulichkeit der

vom Clientsystem gesendeten Nachricht schützt und die Nachricht an den MTA übermittelt.

Sobald die Nachricht komplett dem MTA übertragen wurde und der MTA das Ankommen der Nachricht bestätigt, übergibt das Clientmodul die Verantwortung für die Nachricht an den MTA. Die Übermittlung von Nachrichten zwischen MTAs ist nicht Bestandteil dieser Spezifikation.

Es liegt in der Verantwortung des Clientmoduls sicher zu stellen, dass die Nachricht erfolgreich dem MTA übertragen wird. Falls die Übermittlung einer Nachricht an den MTA fehlschlägt (z.B. bei Verbindungsaufbau mit dem MTA, Authentifizierung gegenüber dem MTA, Verschlüsselung oder Signieren der Nachricht), benachrichtigt das Clientmodul das Clientsystem unter Verwendung entsprechenden SMTP-Antwortcodes über den Fehler.

Beispiel: Verwendet das Clientsystem beim Senden von Nachrichten falsche Anmeldungsdaten, erhält es vom Clientmodul „535 5.7.8 Der Nutzer konnte nicht authentifiziert werden“ als Antwort auf sein AUTH-Kommando.

Das Verhalten des Clientmoduls beim Senden von Nachrichten wird mit Hilfe der in Abbildung 5 dargestellten Zustandsmuster beschrieben werden. Die im Dokument dargestellten Zustände haben nur illustrativen und keinen normativen Charakter. Die Umsetzung kann sich unterscheiden, solange das Ergebnis das Gleiche ist. Die den Zuständen zugeordnete Anforderungen sind normativ, können aber außerhalb des Kontexts dieser Zustände umgesetzt werden.

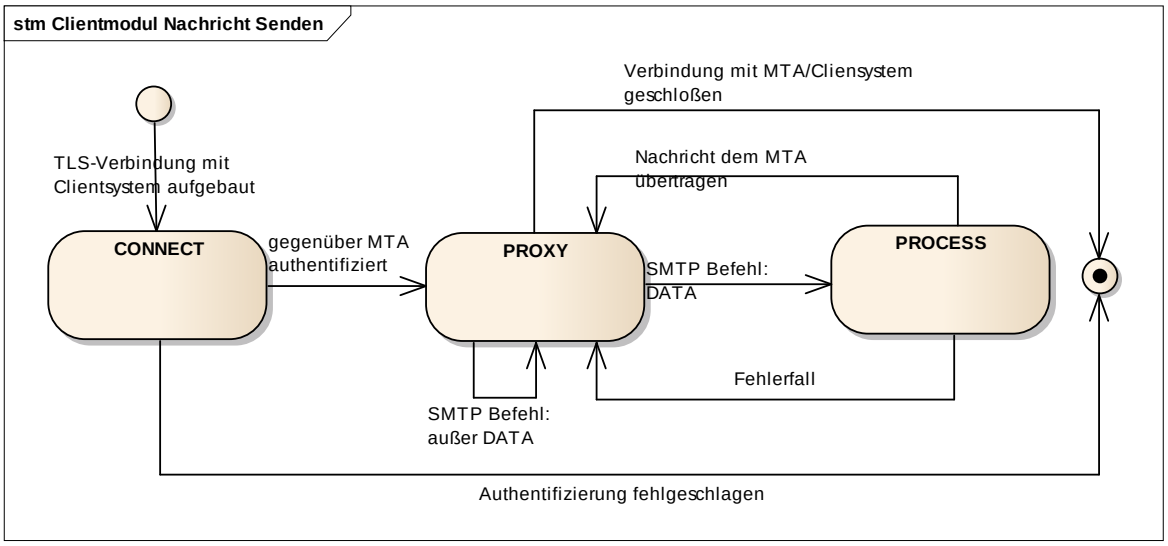

Abbildung 5: Abb_State_CM_Send Zustände Clientmodul beim Senden von Nachrichten

Das Clientmodul lauscht auf einem TCP Port und wartet bis ein Clientsystem mit ihm eine Verbindung aufbaut. Sobald dies passiert, geht das Clientmodul in den CONNECT-Zustand über und betrachtet die SMTP-Verbindung als geöffnet. Die Verbindung zwischen dem Clientsystem und dem Clientmodul muss mit TLS geschützt werden. Um mit marktüblichen E-Mail-Clients kompatibel zu sein ist die Unterstützung von TLS 1.0 obligatorisch.

Im CONNECT-Zustand führt das Clientmodul einen SMTP-Dialog mit dem Clientsystem, in dem ihm die Anmeldedaten des Nutzers sowie die Adresse und die Portnummer des MTAs mitgeteilt werden. Sobald die Anmeldedaten und die Adresse des MTAs übermittelt

sind, baut das Clientmodul eine über TLS geschützte SMTP-Verbindung mit dem MTA auf, authentifiziert sich und geht in den PROXY-Zustand über.

Im PROXY-Zustand leitet das Clientmodul SMTP-Kommandos und SMTP-Antwortcodes zwischen dem Clientsystem und dem MTA weiter, bis das Clientsystem mit dem DATA-Kommando die Übertragung einer Nachricht initiiert. Sobald das Clientsystem anfängt, Inhalte einer Nachricht zu übertragen, geht das Clientmodul in den PROCESS-Zustand über.

In PROCESS-Zustand wird die Nachricht entsprechend dem KOM-LE-S/MIME-Profile [gemSMIME_KOMLE] geschützt und anschließend an den MTA übermittelt. Sobald die Nachricht erfolgreich an den MTA übertragen wurde oder im Fehlerfall, geht das Clientmodul in den PROXY-Zustand zurück.

Nachdem die Verbindungen zwischen dem Clientsystem, dem Clientmodul und dem MTA aufgebaut wurden, übermittelt das Clientmodul die SMTP-Meldungen zwischen dem Clientsystem und dem MTA so lange die beiden Verbindungen bestehen.

3.2.2 CONNECT-Zustand

Sobald die TCP-Verbindung zwischen dem Clientsystem und dem Clientmodul aufgebaut ist, geht das Clientmodul in den CONNECT-Zustand über.

3.2.2.1 Initialisierung

⌦ **KOM-LE-A_2007 SMTP Begrüßung**

Nachdem die SMTP-Verbindung zwischen dem Clientsystem und dem Clientmodul aufgebaut ist, MUSS das Clientmodul dem Clientsystem die SMTP-Begrüßung senden. Um zu signalisieren, dass Extended SMTP unterstützt wird, muss die Begrüßung „ESMTP“ enthalten. ⌫

Beispiel einer solchen Begrüßung: 220 KOM-LE-Clientmodul ESMTP

Das Clientmodul führt einen SMTP-Dialog mit dem Clientsystem bis zum Punkt, an dem das Clientsystem ihm die Adresse und die Portnummer des MTAs als einen Teil des während des Authentifizierungsverfahrens übertragenen Benutzernamens mitteilt (siehe Kapitel 3.2.2.2).

Tabelle 1 beschreibt Antworten, die das Clientmodul dem Clientsystem im CONNECT-Zustand sendet.

Tabelle 1: Tab_SMTP_Ant_Init Antworten Clientmodul im CONNECT-Zustand

SMTP-Kommando (Clientsystem -> Clientmodul)	**SMTP-Antwortcode (Clientmodul -> Clientsystem)**
HELO	“250 OK” Antwortcode
EHLO	“250 OK” Antwortcode mit folgenden EHLO Kennworten: SIZE <size> AUTH LOGIN PLAIN

SMTP-Kommando (Clientsystem -> Clientmodul)	SMTP-Antwortcode (Clientmodul -> Clientsystem)
	8BITMIME ENCHANCEDSTATUSCODES und <size> gleich oder großer als 35882577
AUTH	Anmeldungsdaten erhalten und Verbindungsaufbau mit dem MTA beginnen (siehe Kapitel 3.2.2.2)
RSET, NOOP	„250 OK“ Antwortcode
MAIL, RCPT, DATA	„530 5.7.0“ Antwortcode (Authentication required)
QUIT	„221 OK“ Antwortcode senden und die Verbindung mit dem Clientsystem schließen
Andere Meldungen	„502 5.5.1“ Antwortcode (Invalid command)

⌦ **KOM-LE-A_2008 Initialer SMTP-Dialog**

Das Clientmodul MUSS, nachdem die SMTP-Verbindung zwischen dem Clientsystem und dem Clientmodul aufgebaut wird und bis zum Punkt an dem das Clientsystem die Bestätigung des Erfolgs oder Misserfolgs seiner Authentifizierung erwartet, einen SMTP-Dialog entsprechend der Tabelle Tab_SMTP_Ant_Init mit dem Clientsystem führen. ⌫

3.2.2.2 Verbindungsaufbau mit MTA

Das Clientmodul kann die Verbindung mit dem MTA nur dann aufbauen, wenn ihm das Clientsystem die Adresse des MTAs und die Portnummer des SMTP-Dienstes übermittelt. Das Clientmodul erwartet, dass ihm der Domain Name oder die IP-Adresse und die Portnummer während des Authentifizierungsverfahrens als Teil des Benutzernamens mitgeteilt werden.

Das Clientmodul führt das Authentifizierungsverfahren mit dem Clientsystem bis zu dem Punkt, an dem es mit dem entsprechenden Antwortcode die Authentifizierung akzeptieren oder ablehnen muss. Das Clientmodul allein kann das Clientsystem nicht authentifizieren. Die Authentizität der Zugangsdaten kann nur vom MTA überprüft werden. Dazu authentifiziert sich das Clientmodul im Auftrag vom Clientsystem gegenüber dem MTA.

Die MTA-Adresse und die Portnummer des SMTP-Dienstes sind als Teil des SMTP-Benutzernamens vom Clientsystem zu übergeben. Sie sind vom eigentlichen Benutzernamen durch das Zeichen ’#’ getrennt und als adresse:port String formatiert.

Um mit der SM-B über den Konnektor kommunizieren zu können, werden dem KOM-LE-Clientmodul ebenfalls als Teil des SMTP-Benutzernamens, die Prameter

- MandantId,
- ClientSystemId und
- WorkplaceId

übergeben (siehe Kapitel 3.5 und [gemSpec_Kon] für Details zu MandantId, ClientSystemId und WorkplaceId). Die Parameter entsprechen denen des aufrufenden Clients und werden voneinander durch das Zeichen '#' getrennt.

Der Aufbau des SMTP-Benutzernamens entspricht somit dem folgenden Muster:

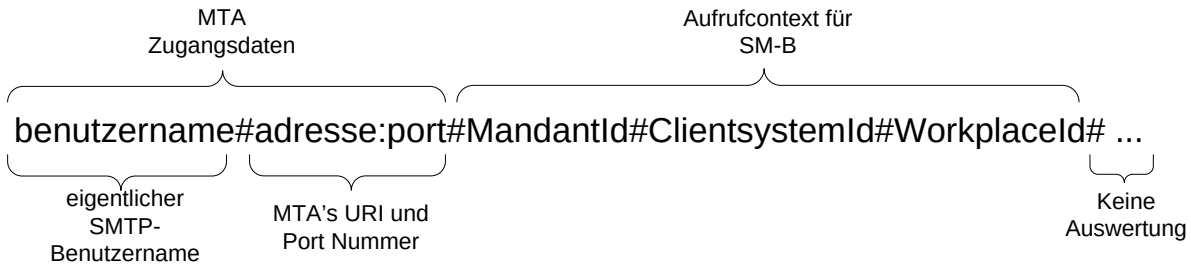

Abbildung 6: Abb_MTA_Nutzername Format des SMTP- Benutzernamens

Beispiel:

Bei folgenden Informationen

- Benutzername des Clients = „erik.mustermann@komle.de",
- Domain Adresse des MTAs = „mail.komle.de" und Portnummer = 465,
- MandantId = 1,
- ClientsystemId = KOM_LE,
- WorkplaceId = 7

erwartet das Clientmodul, dass das Clientsystem ihm folgenden SMTP-Benutzernamen als String überträgt:

```
erik.mustermann@komle.de#mail.komle.de:465#1#KOM_LE#7
```

Das KOM-LE-Clientmodul bricht die Kommunikation mit dem entsprechende SMTP-Antwortcode ab (siehe Tabelle 2), wenn der erhaltene SMTP-Benutzername nicht alle erforderlichen Parameter enthält. Beinhaltet der SMTP-Benutzername zusätzliche durch ‚#' abgegrenzte Parameter (z.B. #UserId), werden diese Parameter vom Clientmodul nicht ausgewertet und der Sendevorgang wird fortgesetzt.

Für SMTP-Authentifizierung existieren sowohl Mechanismen für die Übertragung von Nutzername und Passwort im Klartext (PLAIN und LOGIN) als auch Challenge-Response-Mechanismen. Die auf Challenge-Response (DIGEST-MD5, CRAM-MD5, NTLM) basierenden Mechanismen machen das Extrahieren des Passworts aus der Challenge-basierten Response für das Clientmodul unmöglich. Deshalb werden für die SMTP-Authentifizierung nur die PLAIN oder LOGIN-Mechanismen verwendet.

Sobald das Clientmodul die Anmeldedaten des Nutzers erhält, extrahiert es die Adresse des MTAs und die Portnummer des SMTP-Dienstes aus dem Nutzernamen und baut damit die Verbindung zum MTA auf. Die Verbindung wird über TLS geschützt. Details zum Aufbau der TLS-Verbindung werden in Kapitel 4.1.3 beschrieben.

Tabelle 2 enthält SMTP-Antwortcodes, die das Clientmodul dem Clientsystem bei einem Verbindungsaufbau mit dem MTA übermittelt.

Tabelle 2: Tab_SMTP_Verbindung SMTP-Antwortcodes für MTA-Verbindungsaufbau

Bedingung	SMTP-Antwortcode (Clientmodul -> Clientsystem)
Das Clientmodul hat sich erfolgreich gegenüber dem MTA mit den vom Clientsystem erhaltenen Anmeldungsdaten authentifiziert.	235 2.7.0 (Authentication successful)
Das Clientsystem verwendet für die SMTP-Authentifizierung einen anderen Mechanismus als PLAIN oder LOGIN.	504 5.7.4 (Security features not supported)
Die vom Clientsystem erhaltene SMTP-Authentifizierungsidentität ist nicht vollständig (MTA-Adresse, MandantId, ClientSystemId oder WorkplaceID fehlt – siehe Abbildung 6)	501 5.5.4 (Invalid command arguments)
Die Verbindung zwischen dem Clientmodul und dem MTA kann nicht aufgebaut werden.	454 4.7.0 (Temporary authentication failure)
Die Authentifizierung gegenüber dem MTA schlägt fehl.	535 5.7.8 (Authentication credentials invalid)

Die Verbindungen zwischen dem Clientsystem und dem Clientmodul sowie zwischen dem Clientmodul und dem MTA bleiben solange offen, bis eine von beiden geschlossen oder abgebrochen wird. Sobald eine der beiden Verbindungen geschlossen oder abgebrochen wird, übermittelt das Clientmodul die ausstehenden SMTP-Meldungen und schließt die andere Verbindung. Die SMTP-Sitzung wird damit für den MTA, das Clientsystem und das Clientmodul beendet.

Beispiel: Nachdem das Clientmodul das QUIT-Kommando vom Clientsystem erhalten und dem MTA übermittelt hat, bestätigt der MTA das Ankommen des Kommandos mit dem „221“ Antwortcode und schließt die Verbindung mit dem Clientmodul. Das Clientmodul übermittelt den „221“ Antwortcode dem Clientsystem und schließt die Verbindung mit dem Clientsystem.

⌦ **KOM-LE-A_2009 Unterstützung der Serverteile der Mechanismen PLAIN und LOGIN**

Das Clientmodul MUSS für die SMTP-Authentifizierung des Clientsystems ausschließlich die Serverteile der SASL-Mechanismen PLAIN und LOGIN unterstützen. ⌫

⌦ **KOM-LE-A_2010 Extrahieren von MTA-Adresse, Portnummer und Kartenaufrufkontext**

Das Clientmodul MUSS den Benutzernamen, die MTA-Adresse, die zugehörige Portnummer und den Kartenaufrufkontext aus dem vom Clientsystem erhaltenen SMTP-Benutzernamen entsprechend Abbildung Abb_MTA_Nutzer_Name extrahieren. ⌫

KOM-LE-A_2011 Verbindungsaufbau mit dem MTA über MTA-Adresse und Portnummer

Das Clientmodul MUSS die MTA-Adresse und die Portnummer, die aus dem vom Clientsystem erhaltenen SMTP-Benutzernamen extrahiert wurden (siehe Abbildung Abb_MTA_Nutzer_Name), für den Verbindungsaufbau mit dem MTA verwenden. ⌫

KOM-LE-A_2012 Authentisierung gegenüber dem MTA mit Benutzernamen und Passwort

Das Clientmodul MUSS den Benutzernamen, der aus dem vom Clientsystem erhaltenen SMTP-Benutzernamen extrahiert wurde (siehe Abbildung Abb_MTA_Nutzer_Name) sowie das vom Clientsystem erhaltene Passwort für die Authentisierung gegenüber den MTA verwenden. ⌫

KOM-LE-A_2013 Unterstützung der Clientteile der Mechanismen PLAIN und LOGIN

Das Clientmodul MUSS für die SMTP-Authentifizierung mit dem MTA die Clientteile der der SASL-Mechanismen PLAIN und LOGIN unterstützen. ⌫

KOM-LE-A_2014 Authentifizierung gegenüber MTA mit anderen Mechanismen als PLAIN und LOGIN

Das Clientmodul KANN für die Authentifizierung gegenüber dem MTA andere Authentifizierungsmechanismen als PLAIN oder LOGIN benutzen. ⌫

KOM-LE-A_2015 Ergebnis des Verbindungsaufbaus mit dem MTA

Das Clientmodul MUSS das Clientsystem über das Ergebnis des Verbindungsaufbaus mit dem MTA mit den in Tabelle Tab_SMTP_Verbindung beschriebenen SMTP-Antwortcodes informieren. ⌫

KOM-LE-A_2016 Schließen der SMTP-Verbindung mit dem Clientsystem

Das Clientmodul MUSS die SMTP-Verbindung mit dem Clientsystem aufrechterhalten. Das Schließen der Verbindung ist nur bei folgenden Ausnahmen zulässig:

- Nachdem die Verbindung zwischen dem Clientmodul und dem MTA geschlossen oder abgebrochen wurde. In diesem Fall MUSS das Clientmodul die Verbindung mit dem Clientsystem schließen. Falls es vom MTA erhaltene und vom Clientsystem noch nicht übertragene SMTP-Antwortcodes gibt, MUSS das Clientmodul diese Antwortcodes an das Clientsystem weiterleiten und danach die Verbindung mit dem Clientsystem schließen.
- Wenn der MTA innerhalb eines konfigurierbaren Timeouts nicht auf ein SMTP-Kommando reagiert. In diesem Fall MUSS das Clientmodul den Antwortcode „421“ an das Clientsystem senden und anschließend die Verbindung schließen.
- Wenn die Verbindung zwischen dem Clientmodul und dem MTA noch nicht aufgebaut wurde und das Clientsystem das QUIT-Kommando übermittelt. In diesem Fall MUSS das Clientmodul mit „221 OK“

Antwortcode antworten und die Verbindung mit dem Clientsystem schließen. ⌫

⌦ **KOM-LE-A_2017 Schließen der SMTP-Verbindung mit dem MTA**

Das Clientmodul MUSS die SMTP-Verbindung mit dem MTA aufrechterhalten. Das Schließen der Verbindung ist nur zulässig:

- Nachdem die Verbindung zwischen dem Clientmodul und dem Clientsystem geschlossen oder abgebrochen wird. In diesem Fall MUSS das Clientmodul die Verbindung mit dem MTA schließen. Falls es vom Clientsystem erhaltene und dem MTA noch nicht übertragene SMTP-Meldungen gibt, MUSS das Clientmodul diese Meldungen dem MTA übertragen, und nur danach die Verbindung mit dem MTA schließen.
- Wenn das Clientmodul innerhalb eines konfigurierbaren Timeouts keine neuen SMTP-Kommandos sendet. In diesem Fall MUSS das Clientmodul die Verbindung mit dem MTA schließen. ⌫

Nachdem sich das Clientsystem gegenüber dem MTA erfolgreich authentifiziert hat, geht das Clientmodul in den PROXY-Zustand über. Anderenfalls bleibt das Clientmodul im CONNECT-Zustand.

3.2.3 PROXY-Zustand

Im PROXY-Zustand vermittelt das Clientmodul SMTP-Meldungen und Antwortcodes zwischen dem Clientsystem und dem MTA. Das Clientmodul bleibt in diesem Zustand bis das Clientmodul das DATA-Kommando bekommt und der MTA das Erhalten von diesem Kommando mit dem Antwortcode „354“ bestätigt. Das Clientmodul leitet den Antwortcode „354“ an das Clientsystem weiter und geht in den PROCESS-Zustand über.

⌦ **KOM-LE-A_2018 Weiterleitung von SMTP-Meldungen und Antwortcodes**

Nach erfolgreicher Beendigung des Authentifizierungsverfahrens mit dem MTA MUSS das Clientmodul alle vom Clientsystem erhaltenen SMTP-Meldungen, mit Ausnahme des RCPT-Kommandos und der Inhalte von E-Mail-Nachrichten sowie alle vom MTA erhaltenen Antwortcodes ohne Veränderung dem MTA bzw. dem Clientsystem unverzüglich übermitteln. ⌫

⌦ **KOM-LE-A_2176 Prüfen auf gültiges ENC-Zertifikat für den Empfänger im RCPT-Kommando**

Das Clientmodul MUSS, wenn es vom Clientsystem ein RCPT TO:<recipient-address> Kommando erhält, prüfen, ob für den im Kommando aufgeführten Empfänger mindestens ein gültiges ENC-Zertifikat existiert. Da die Nachricht nur an Empfänger, die ein gültiges ENC-Zertifikat besitzen weitergeleitet werden darf, MUSS dass Clientmodul im Negativfall das Kommando verwerfen. Im Positivfall MUSS das Clientmodul das Kommando an den MTA weiterleiten. ⌫

3.2.4 PROCESS-Zustand

Im PROZESS-Zustand nimmt das Clientmodul die Inhalte der vom Clientsystem gesendeten Nachricht entgegen. Mit Hilfe von Diensten der TI-Plattform schützt es die

Vertraulichkeit und Integrität der Nachricht entsprechend dem KOM-LE-S/MIME-Profil [gemSMIME_KOMLE]. Anschließend leitet das Clientmodul die geschützte Nachricht an den MTA, bei dem der Nutzer registriert ist, weiter. Im Erfolgsfall wird das Clientsystem über das Versenden der Nachricht informiert. Im Fehlerfall wird das Clientsystem mit dem entsprechenden Antwortcode über den Fehler benachrichtigt. Im folgenden Text wird eine entsprechend dem KOM-LE-S/MIME-Profil geschützte Nachricht auch als KOM-LE-S/MIME-Nachricht bezeichnet.

3.2.4.1 Empfang und Weiterleitung einer Nachricht

Nachdem die Bereitschaft zum Empfangen der Nachricht dem Clientsystem mit dem Antwortcode „354“ bestätigt wurde, erwartet das Clientmodul, dass das Clientsystem mit der Übertragung der Nachricht fortfährt. Die Inhalte der Nachricht werden im Clientmodul zwischengespeichert und sobald das Clientsystem durch die „<CRLF>.<CRLF>“ Zeichensequenz das Ende der Nachricht markiert, werden die Inhalte der Nachricht im Clientmodul durch digitale Signatur und die Verschlüsselung geschützt. Die Details werden im Kapitel 3.2.4.1.1 beschrieben.

KOM-LE bietet die Möglichkeit Nachrichten, die beim Abholen nicht entschlüsselt wurden (z.B. auf Grund eines fehlenden HBA mit dem entsprechenden privaten Schlüssel), nachträglich zu entschlüsseln. Um die nachträgliche Entschlüsselung einer verschlüsselten KOM-LE-Nachricht durchführen zu können, schickt der Empfänger die verschlüsselte Nachricht als ein `message/rfc822` Anhang in einer neuen Nachricht an seine eigene E-Mail-Adresse. Beim nächsten Abholvorgang kann diese Nachricht, sofern die erforderliche Karte vorhanden ist, durch das Clientmodul entschlüsselt werden. Werden solche Nachrichten im Clientmodul erkannt, werden sie weder signiert noch verschlüsselt. Stattdessen wird die verschlüsselte KOM-LE-Nachricht aus dem `message/rfc822` Anhang extrahiert und die `from` Header-Elemente werden durch das `from` Header-Element (E-Mail-Adresse des Absenders) der angekommenen `multipart` MIME-Nachricht ersetzt. Anschließend wird die Nachricht dem MTA übermittelt. Die Details werden im Kapitel 3.2.4.1.2 beschrieben.

Die Benachrichtigung des Clientsystems über den Erfolg des Sendens einer Nachricht findet nur dann statt, wenn der MTA die Übernahme der Verantwortung für die Nachricht mit positiven Erledigungsstatus über den „250“ Antwortcode bestätigt. Ab diesem Moment gilt die Nachricht für das Clientsystem als versendet und der MTA hat sich zu ihrer Lieferung oder Benachrichtigung des Senders über einen Fehlerfall verpflichtet.

Nachdem das Clientsystem über das erfolgreiche Senden der Nachricht oder über einen Fehlerfall mit entsprechendem Antwortcode benachrichtigt wurde, löscht das Clientmodul die zwischengespeicherten Inhalte der Nachricht und geht zurück in den PROXY-Zustand.

⌦ **KOM-LE-A_2019 Signatur und Verschlüsselung entsprechend KOM-LE-S/MiME-Profil**

Das Clientmodul MUSS die vom Clientsystem erhaltene KOM-LE-Nachricht entsprechend dem KOM-LE-S/MIME-Profil [gemSMIME_KOMLE] signieren und verschlüsseln und anschließend dem MTA übermitteln. ⌫

3.2.4.1.1 Bearbeitung einer ungeschützten Nachricht

Um die Vertraulichkeit und die Integrität einer Nachricht zu schützen wird die Nachricht

entsprechend dem KOM-LE-S/MIME-Profil signiert und verschlüsselt. Für das Signieren und die Verschlüsselung nutzt das Clientmodul die Dienste der TI-Plattform. Die folgende Abbildung stellt den prinzipiellen Ablauf und die Aktivitäten des Clientmoduls beim Erzeugen einer dem KOM-LE-S/MIME-Profil entsprechenden Nachricht dar.

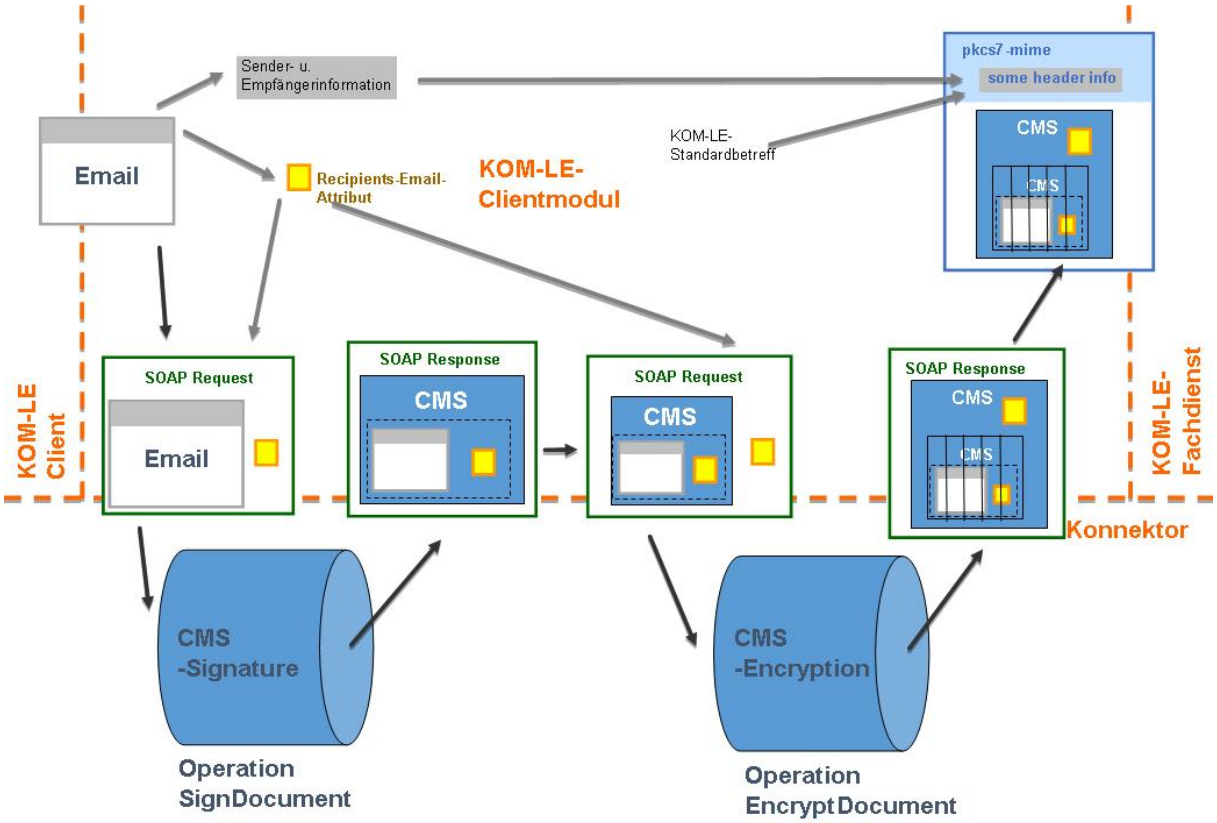

Abbildung 7: Abb_Sig_Verschl Signieren und Verschlüsseln entsprechend S/MIME Profil

Für das digitale Signieren einer Nachricht verwendet das Clientmodul den privaten PrK.HCI.OSIG-Schlüssel der SM-B. Der Zugriff auf die entsprechende Karte und die Erstellung der Signatur erfolgt über die Aufrufe der entsprechenden Operationen der Außenschnittstelle des Konnektors. Eine detaillierte Beschreibung erfolgt im Kapitel 3.5.1.

Wenn das Signieren fehlschlägt, wird das Senden der Nachricht abgebrochen indem dem MTA das RSET-Kommando übermittelt wird und das Clientsystem mit dem Antwortcode „451“ inklusive der entsprechenden Fehlermeldung über den Fehlerfall informiert wird.

⌦ **KOM-LE-A_2177 Verwenden von SignDocument und EncryptDocument**

Das Clientmodul MUSS für das Signieren und Verschlüsseln der Nachrichten die Operationen SignDocument und EncryptDocument der Außenschnittstelle des Konnektors verwenden. ⌫

⌦ **KOM-LE-A_2190 Übergabe des recipient-emails Attributs beim Signieren**

Das Clientmodul MUSS beim Aufruf der Operation SignDocument des Konnektors das recipient-emails Attribut als Aufrufparameter in der ASN.1-Form

```
Attribute ::= SEQUENCE {
        attrType OBJECT IDENTIFIER,
```

```
            attrValues SET OF AttributeValue }
```

übergeben. Das ASN.1-Atribut MUSS DER-kodiert und base64 verpackt im Request-Element

<SIG:SignDocument>/<SIG:SignRequest>/<SIG:OptionalInputs>/<dss:Properties>/<dss:SignedProperties>/<dss:Property>/<dss:Value>/<CMSAttribute>

übergeben werden. ⌫

Folgend ein Beispiel für den SOAP-Request beim Signieren:


```
<?xml version="1.0" encoding="UTF-8" ?>
<SIG:SignDocument
xmlns:CERTCMN="http://ws.gematik.de/conn/CertificateServiceCommon/v2.0"
xmlns:CONN="http://ws.gematik.de/conn/ConnectorCommon/v5.0"
xmlns:CCTX="http://ws.gematik.de/conn/ConnectorContext/v2.0"
xmlns:SIG="http://ws.gematik.de/conn/SignatureService/v7.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema">
  <CONN:CardHandle>zDgq6V5EsA</CONN:CardHandle>
  <CCTX:Context>
  <CONN:MandantId>Praxis Dr. Mustermann</CONN:MandantId>
  <CONN:ClientSystemId>Mediakom-PVS-3000</CONN:ClientSystemId>
  <CONN:WorkplaceId>Arztzimmer2</CONN:WorkplaceId>
  </CCTX:Context>
  <SIG:TvMode>NONE</SIG:TvMode>
  <SIG:SignRequest RequestID="SignRequestNo_001">
  <SIG:OptionalInputs>
  <SIG:KeyReference>C.OSIG</SIG:KeyReference>
  <dss:SignatureType>urn:ietf:rfc:5652</dss:SignatureType>
  <dss:Properties>
  <dss:SignedProperties>
  <dss:Property>
  <dss:Identifier>RecipientEmailsAttribute</dss:Identifier>
  <dss:Value>
<CMSAttribute>QnNVakJzUjA5RWJHaGpaMGRUUVV4TlVqQnNSMDlFYkdoalowZFRRVXhOUVVGQ
lVRVUZCVVVOQlJVMXRRMXAwZFUxR1VYaEVVemhp</CMSAttribute>
  </dss:Value>
  </dss:Property>
  </dss:SignedProperties>
  </dss:Properties>
  <SIG:IncludeEContent>true</SIG:IncludeEContent>
  </SIG:OptionalInputs>
  <SIG:Document ShortText="none">
<dss:Base64Data>TUlNRS1WZXJzaW9uOiAxLjANCkNvbnRlbnQtdHlwZTogdGV4dC9wbGFpbjs
gY2hhcnNldD1pc28tODg1OS0xNQ0KQ29udGVudC1UcmFuc2Zlci1FbmNvZGluZzogOGJpdA0KRn
JvbTogPGhhbnMubXVzdGVyYXJ6dEBwcmF4aXNBLmRlPg0KVG86IDxldmEubXVzdGVyYXJ6dEBwc
mF4aXNCLmRlPg0KU3ViamVjdDog3GJlcndlaXN1bmcgSHIuIE0uIFBhdGllbnRCDQpEYXRlOiBN
b24sIDExIE5vdiAyMDEzIDE0OjM0OjI3ICswMTAwDQoNClNlaHIgZ2VlaHJ0ZSBGcmF1IEtvbGx
lZ2luIERyLiBNdXN0ZXJhcnp0LA0KDQpoaWVybWl0IPxiZXJ3ZWlzZSBpY2ggSWhuZW4gSHIuIE
0uIFBhdGllbnRCIGF1ZiBHcnVuZCAuLi4uDQoNCk1pdCBmcmV1bmRsaWNoZW4gR3L832VuLA0KD
QpEci4gSGFucyBNdXN0ZXJhcnp0</dss:Base64Data>
  </SIG:Document>
  <SIG:IncludeRevocationInfo>false</SIG:IncludeRevocationInfo>
  </SIG:SignRequest>
 </SIG:SignDocument>
```


⌦ KOM-LE-A_2020 Signieren der Nachricht mit dem Schlüssel Prk.HCI.OSIG

Das Clientmodul MUSS für das Signieren einer KOM-LE-Nachricht den privaten Schlüssel PrK.HCI.OSIG der SM-B der medizinischen Institution verwenden. ⌫

⌦ **KOM-LE-A_2021 Verhalten, wenn Nachricht nicht signiert werden kann**

Das Clientmodul MUSS dem MTA das Kommando RSET senden und das Clientsystem mit dem Antwortcode „451“ benachrichtigen, wenn das Clientmodul die vom Clientsystem erhaltene Nachricht nicht digital signieren kann. ⌫

Die Verschlüsselung erfolgt sowohl für den Sender als auch für alle Empfänger. Die erforderlichen Verschlüsselungszertifikate C.HCI.ENC für Institutionen und C.HP.ENC für Leistungserbringer werden im Verzeichnisdienst zur Verfügung gestellt. Für die Suche nach den passenden Einträgen im Verzeichnisdienst wird die KOM-LE-E-Mail-Adresse als Suchschlüssel verwendet. Wenn der Sender bzw. ein Empfänger mehrere Verschlüsslungszertifikate hat (z.B. wenn dem Empfänger ein neuer HBA ausgegeben wurde und der alte noch gültig ist), wird die Nachricht mit allen vorhandenen Verschlüsselungszertifikaten verschlüsselt.

⌦ **KOM-LE-A_2191 Übergabe des recipient-emails Attributs beim Verschlülsseln**

Das Clientmodul MUSS beim Aufruf der Operation EncryptDocument des Konnektors das recipient-emails Attribut als Aufrufparameter in der ASN.1-Form

```
Attribute ::= SEQUENCE {
        attrType OBJECT IDENTIFIER,
        attrValues SET OF AttributeValue }
```

übergeben. Das ASN.1-Atribut MUSS DER-kodiert und base64 verpackt im Request-Element

<CRYPT:EncryptDocument>/<CRYPT:OptionalInputs>/<CRYPT:UnprotectedProperties>/<dss:Property>/<dss:Value>/<CMSAttribulte>

übergeben werden. ⌫

Folgend ein Beispiel für den SOAP-Request beim Verschlüsseln:

```
<?xml version="1.0" encoding="UTF-8" ?>
<CRYPT:EncryptDocument
xmlns:CONN="http://ws.gematik.de/conn/ConnectorCommon/v5.0"
xmlns:CCTX="http://ws.gematik.de/conn/ConnectorContext/v2.0"
xmlns:CRYPT="http://ws.gematik.de/conn/EncryptionService/v6.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema">
  <CCTX:Context>
  <CONN:MandantId>Praxis Dr. Mustermann</CONN:MandantId>
  <CONN:ClientSystemId>Mediakom-PVS-3000</CONN:ClientSystemId>
  <CONN:WorkplaceId>Arztzimmer2</CONN:WorkplaceId>
  </CCTX:Context>
  <CRYPT:RecipientKeys>
  <CRYPT:CertificateOnCard>
  <CONN:CardHandle>zDgq6V5EsA</CONN:CardHandle>
  <CRYPT:KeyReference>C.ENC1.R2048</CRYPT:KeyReference>
  </CRYPT:CertificateOnCard>
<CRYPT:Certificate>UjBsR09EbGhjZ0dTQUxNQUFBUUNBRU1tQ1p0dU1GUXhEUzhi</CR
YPT:Certificate>
  </CRYPT:RecipientKeys>
  <CONN:Document>
<dss:Base64Data>QnNVakJzUjA5RWJHaGpaMGRUUVV4TlVqQnNSMDlFYkdoalowZFRRVXh
OUVVGQlVRVUZCVVVOQlJVMXRRMXAwZFUxR1VYaEVVemhp</dss:Base64Data>
  </CONN:Document>
  <CRYPT:OptionalInputs>
  <CRYPT:EncryptionType>urn:ietf:rfc:5652</CRYPT:EncryptionType>
  <CRYPT:UnprotectedProperties>
```

```
    <dss:Property>
    <dss:Identifier>RecipientEmailsAttribute</dss:Identifier>
    <dss:Value>
  <CMSAttribute>QnNVakJzUjA5RWJHaGpaMGRUUVV4TlVqQnNSMDlFYkdoalowZFRRVXhOU
  VVGQlVRVUZCVVVOQlJVMXRRMXAwZFUxR1VYaEVVemhp</CMSAttribute>
    </dss:Value>
    </dss:Property>
    </CRYPT:UnprotectedProperties>
    </CRYPT:OptionalInputs>
  </CRYPT:EncryptDocument>
```

⌦ **KOM-LE-A_2022 Verschlüsseln der Nachricht mit den Verschlüsselungszertifikaten C.HCI.ENC bzw. C.HP.ENC**

Das Clientmodul MUSS vom Clientsystem erhaltene E-Mail-Nachrichten sowohl für jeden in den RCPT-Kommandos angegeben Empfänger als auch für den Sender aus dem `from` bzw. `sender` Header-Element der Nachricht mit allen dem Sender bzw. Empfängern zugeordneten Verschlüsselungszertifikaten (C.HCI.ENC für eine Institution oder C.HP.ENC für einen Leistungserbringer) verschlüsseln. ⌫

⌦ **KOM-LE-A_2178 Kein Versenden an Empfänger mit unterschiedlichen Telematik-IDs in den Verschlüsselungszertifikaten**

Existieren für einen Empfänger mehrere Verschlüsselungszertifikate mit unterschiedlichen Telematik-IDs DARF das Clientmodul die Nachricht NICHT an diesen Empfänger versenden. ⌫

⌦ **KOM-LE-A_2192 Fehlernachricht bei Empfänger mit unterschiedlichen Telematik-IDs in den Verschlüsselungszertifikaten**

Existieren für einen Empfänger mehrere Verschlüsselungszertifikate mit unterschiedlichen Telematik-IDs MUSS das Clientmodul den Absender der Nachricht mit einer Fehlernachricht, die weder zu signieren noch zu verschlüsseln ist, informieren. ⌫

⌦ **KOM-LE-A_2023 Verschlüsselungszertifikate aus dem Verzeichnisdienst**

Das Clientmodul MUSS in der Lage sein, die Verschlüsselungszertifikate aus dem Verzeichnisdienst der TI mit Hilfe der E-Mail-Adresse zu ermitteln. ⌫

Nachdem die Nachricht erfolgreich signiert wurde und die entsprechenden Verschlüsselungszertifikate zur Verfügung stehen, führt das Clientmodul die Verschlüsselung der Nachricht für alle Empfänger bzw. Sender durch. Die Empfänger werden über die E-Mail-Adressen aus den RCPT-Kommandos identifiziert. Die Sender werden über die E-Mail-Adressen im `sender` Header-Element identifiziert. Wenn der Header der Nachricht kein `sender` Element enthält, werden die E-Mail-Adressen des Senders aus dem `from` Header-Element übernommen.

Beim Verschlüsselungsvorgang sind die folgenden Szenarien möglich:

- Die Nachricht kann für alle E-Mail-Adressen (sowohl Sender als auch Empfänger) verschlüsselt werden.
- Es gibt E-Mail-Adressen, für die aufgrund der fehlenden oder nicht gültigen Zertifikate die Nachricht nicht verschlüsselt werden kann. In diesem Fall wird die Nachricht mit den verfügbaren Zertifikaten verschlüsselt und an den MTA

übermittelt. Die E-Mail-Adressen für die die Verschlüsselung nicht durchgeführt werden konnte werden aus dem Header entfernt. Der Absender der Nachricht wird über eine im Clientmodul generierte und an den MTA übermittelte E-Mail über den Fehlerfall informiert. Die Nachricht mit der Fehlermeldung wird weder signiert noch verschlüsselt.

- Wenn die Verschlüsselung für keinen der Empfänger durchgeführt werden kann, wird das Senden der Nachricht abgebrochen. Dabei wird dem MTA das RSET-Kommando gesendet und das Clientsystem wird mit dem Antwortcode „451“ und der entsprechenden Fehlermeldung über den Fehlerfall informiert.

Die Verschlüsselung erfolgt über die Aufrufe der entsprechenden Operationen der Außenschnittstelle des Konnektors. Eine detaillierte Beschreibung erfolgt in Kapitel 3.5.3.

⌦ **KOM-LE-A_2024 Information des Absenders über Empfänger, für die nicht verschlüsselt werden kann**

Kann eine Nachricht auf Grund von fehlenden oder ungültigen Zertifikaten nicht für alle Empfänger verschlüsselt werden, MUSS das Clientmodul den Absender mit einer E-Mail über den Fehlerfall informieren. Aus dem Inhalt der Fehlernachricht müssen alle Empfänger, für die nicht verschlüsselt werden konnte, hervorgehen. Die Fehlernachricht ist weder zu signieren noch zu verschlüsseln. Die Originalnachricht darf an die Empfänger, für die nicht verschlüsselt werden konnte, nicht versendet werden. ⌫

⌦ **KOM-LE-A_2025 Abbruch des Sendens, wenn keine Verschlüsselung möglich**

Das Clientmodul MUSS das Clientsystem mit dem Antwortcode „451“ benachrichtigen und den Senden-Vorgang zum MTA mit dem RSET-Kommando abbrechen, wenn das Clientmodul die vom Clientsystem erhaltene Nachricht für keinen Empfänger verschlüsseln kann. ⌫

Das KOM-LE-S/MIME-Profil fordert, dass jede entsprechend dem Profil verschlüsselte Nachricht das recipient-emails Attribut enthält. In diesem Attribut werden Zusammenhänge zwischen den für die Verschlüsselung verwendeten Zertifikaten und den E-Mail-Adressen der Empfänger bzw. des Senders angegeben. Das Clientmodul befüllt dieses Attribut nur mit den E-Mail-Adressen für die die Nachricht erfolgreich verschlüsselt werden konnte.

Um die Anzahl von Anfragen an den Verzeichnisdienst und die Bearbeitungszeiten zu reduzieren werden die für die Verschlüsselung verwendeten Zertifikate für eine konfigurierbare Zeitdauer im Clientmodul gecached.

⌦ **KOM-LE-A_2026 Cachen von Verschlüsselungszertifikaten**

Das Clientmodul MUSS das manipulationssichere Cachen von Verschlüsselungszertifikaten für eine konfigurierbare Zeitdauer unterstützen. ⌫

Die folgenden Schritte stellen den Schutzvorgang für eine Nachricht im Clientmodul dar. Die Schritte haben einen beschreibenden und nicht normativen Charakter. Die Umsetzung kann sich unterscheiden, solange die Anforderungen des Dokuments erfüllt sind.

1. Der Cache und anschließend falls erforderlich der Verzeichnisdienst werden für Verschlüsselungszertifikate der Empfänger und Sender durchgesucht. Die

entsprechenden E-Mail-Adressen dienen als die Suchschlüssel.

2. Der Signaturdienst der TI-Plattform wird mit der zu sendenden Nachricht und der Referenz auf den Signaturschlüssel als Aufrufparameter aufgerufen.

3. Der Verschlüsselungsdienst der TI-Plattform wird mit der signierten Nachricht und den gefundenen Verschlüsselungszertifikaten als Aufrufparameter aufgerufen.

4. Die TI-Plattform prüft den Sperrstatus der übergebenen Verschlüsselungszertifikate und führt die Verschlüsselung durch, wenn alle Zertifikate gültig sind. Sollte die Prüfung eines oder mehreren Zertifikate als nicht gültig ausweisen, bricht die TI-Plattform den Verschlüsselungsvorgang ab. Falls sich unter den ungültigen Zertifikaten die aus dem Cache geholten Zertifikate befinden, wird der Verzeichnisdienst nach Ersatzzertifikaten durchsucht.

    a. Falls Ersatzzertifikate gefunden werden, wird der Verschlüsselungsvorgang wiederholt.

    b. Werden keine Ersatzzertifikate gefunden, werden diesen Zertifikaten entsprechende Empfänger aus dem Header der Nachricht entfernt und über den Fehlerfall mit Hilfe einer im Clientmodul generierten E-Mail informiert. Die ursprüngliche Nachricht wird an diese Empfänger nicht gesendet, weil sie nicht in der Lage sind, diese Nachricht zu entschlüsseln.

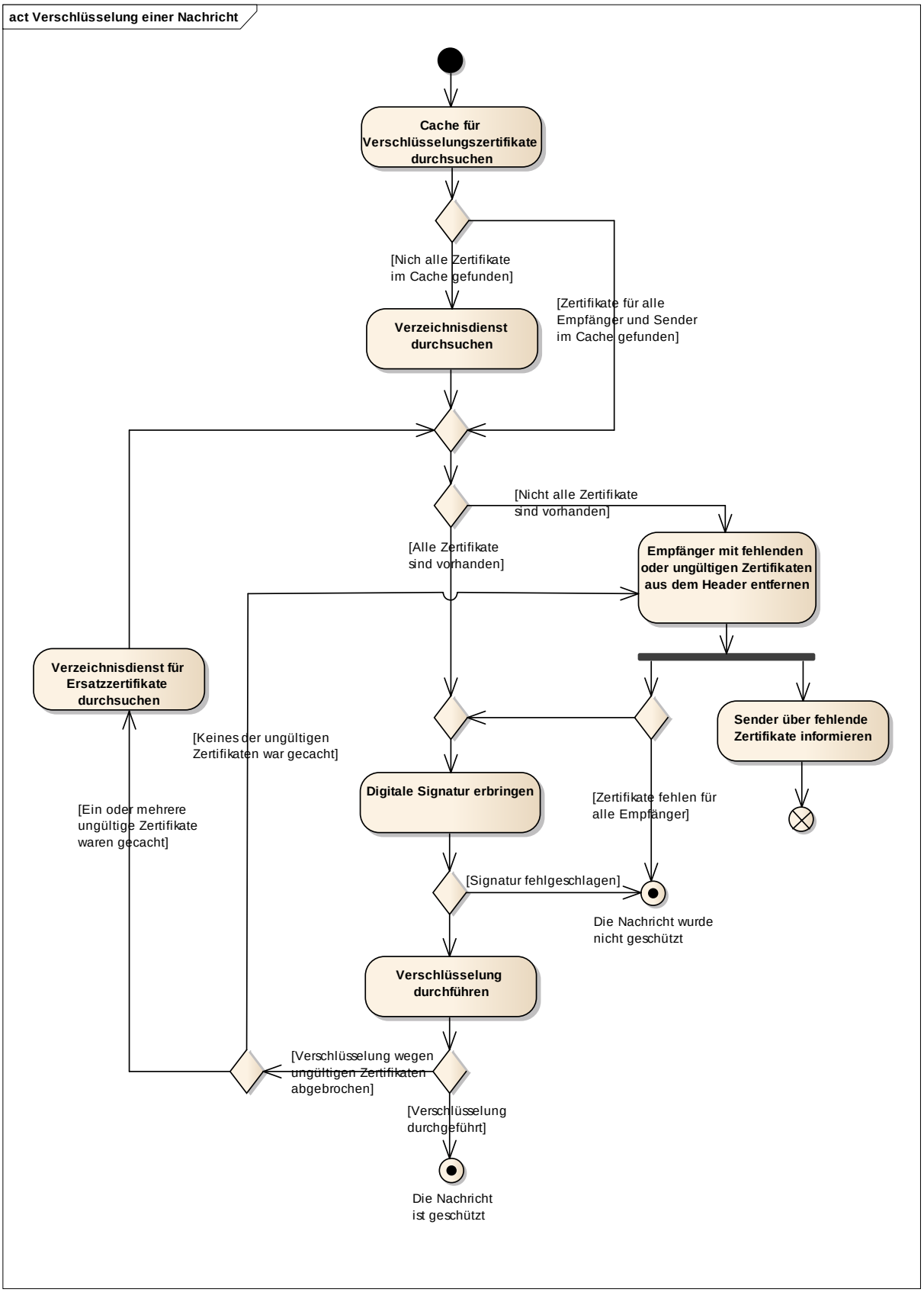

Abbildung 8: Abb_Verschl_Msg Verschlüsselung einer Nachricht

Abbildung 8 stellt die oben beschriebenen Schritte als Aktivitätsdiagramm dar.

⌦ **KOM-LE-A_2027 Befüllung des recipient-emails Attributs**

Das Clientmodul MUSS für die E-Mail-Adressen für die die Nachricht erfolgreich verschlüsselt werden konnte einen Wert in das recipient-emails Attribut entsprechend dem KOM-LE-S/MIME-Profil einfügen. ⌫

⌦ **KOM-LE-A_2028 Entfernen von Empfängern aus dem Header der Nachricht**

Das Clientmodul MUSS die Empfänger bzw. Sender für die die Verschlüsselung der Nachricht nicht durchgeführt werden konnte, aus to, cc bzw. from, sender Header-Elementen der Nachricht entfernen, um sicherzustellen, dass die ursprüngliche Nachricht nicht an solche Empfänger gesendet wird. ⌫

Nachdem die Verschlüsselung durchgeführt wurde, verpackt das Clientmodul das vom Konnektor verschlüsselte CMS-Objekt in eine äußere Nachricht entsprechend KOM-LE-S/MIME-Profil und überträgt die geschützte Nachricht an den MTA.

⌦ **KOM-LE-A_2193 Verpacken des verschlüsselten CMS-Objektes**

Das Clientmodul MUSS das vom Konnektor signierte und verschlüsselte CMS-Objekt in eine äußere Nachricht entsprechend den Anforderungen KOM-LE-A_2097, KOM-LE-A_2098, KOM-LE-A_2099, KOM-LE-A_2100, KOM-LE-A_2101, KOM-LE-A_2102 des KOM-LE S/MIME Profils verpacken. ⌫

3.2.4.1.2 Bearbeitung einer geschützten KOM-LE-Nachricht

Wenn während eines Abholvorgangs eine KOM-LE-Nachricht nicht im Clientmodul entschlüsselt werden konnte, wird sie dem Clientsystem als eine `message/rfc822` Einheit mit einem Fehlertext geliefert (siehe das Beispiel im Kapitel 3.3.4.2.1). Um die Nachricht im Anhang nachträglich zu entschlüsseln und ihre Signatur prüfen zu können, muss der Nutzer die erhaltene Nachricht an seine eigene E-Mail-Adresse senden. Beim nächsten Abholvorgang wird diese Nachricht dann nochmalig im Clientmodul aufbereitet.

⌦ **KOM-LE-A_2029 Aufbereitung einer vom Clientsystem erhaltenen KOM-LE-S/MIME-Nachricht**

Das Clientmodul MUSS die vom Clientsystem empfangene Nachricht, deren Body eine `message/rfc822` MIME Einheit mit einer dem KOM-LE-Profil entsprechenden Nachricht (KOM-LE-S/MIME-Nachricht) enthält, in den folgenden Schritten aufbereiten:

1. Die in `message/rfc822` Einheit enthaltene KOM-LE-S/MIME-Nachricht wird aus der erhaltenen Nachricht extrahiert und dem MTA übergeben.
2. Die vom Clientsystem erhaltene Nachricht wird verworfen. ⌫

Beispiel für die oben beschriebene Transformation:

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="unique-boundary-1"
Subject: WG: Signed and encrypted in attachment
Date: Fri, 10 Feb 2012 14:29:21 +0100
From: musterfrau@komle.de
```

```
To: musterfrau@komle.de

This is a multi-part message in MIME format.

--unique-boundary-1
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

Der f=FCr die Entschl=FCsselung der Nachricht ben=F6tigte Schl=FCssel =
wurde nicht gefunden. =DCberpr=FCfen Sie ob die entsprechende Karte =
gesteckt ist und leiten Sie diese Nachricht an Ihre eigene Email Adresse =
(musterfrau@komle.de) weiter. Beim n=E4chsten Abholen der Nachricht =
wird der Verschl=FCsselungsvorgang wiederholt.

--unique-boundary-1
Content-Type: message/rfc822

X-KOM-LE-Version: 1.0
MIME-Version: 1.0
Content-Type: application/pkcs7-mime; smime-type=enveloped-
data;name="smime.p7m";
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7m"
Subject: Verschlüsselte KOM-LE Nachricht
Date: Fri, 9 Feb 2012 12:07:17 +0100
From: mustermann@komle.de
To: musterfrau@komle.de
Cc: mustermann2@komle.de

<verschlüsselter Inhalt>

--unique-boundary-1
```

Im Clientmodul wird diese Nachricht entsprechend der Anforderung [KOM-LE-A_2029] aufbereitet:

```
X-KOM-LE-Version: 1.0
MIME-Version: 1.0
Content-Type: application/pkcs7-mime;
smime-type=enveloped-data; name="smime.p7m"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7m"
Subject: Verschlüsselte KOM-LE Nachricht
Date: Fri, 9 Feb 2012 12:07:17 +0100
From: mustermann@komle.de
To: musterfrau@komle.de
Cc: mustermann2@komle.de


<Verschlüsselter Inhalt>
```

3.2.5 Beispiele

Das Clientsystem (C) verbindet sich mit dem Clientmodul (M) und sendet dem MTA-Server (S) eine Nachricht (im Beispiel werden auch die Zustände des Clientmoduls dargestellt):

```
C:     <das Clientsystem öffnet eine mit TLS geschützte Verbindung mit dem
Clientmodul>
M:     <CONNECT Zustand>
M->C: 220 KOM-LE Clientmodul ESMTP
```

```
C->M: EHLO [192.168.1.5]
M->C: 250 – SIZE 35882577
M->C: 250 - AUTH LOGIN PLAIN
M->C: 250 – 8BITMIME
M->C: 250 ENCHANCEDSTATUSCODES
C->M: AUTH LOGIN
M->C: 334 VXNlcm5hbWU6
C->M: bXVzdGVybWFubkBrb21sZS5kZSNtYWlsLmtvbWxlLmRlOjU4NyMxI0tPTS1MRSM3==
M->C: 334 UGFzc3dvcmQ6
C->M: lkajsdfvlj
M:     <das Clientmodul öffnet eine mit TLS geschützte Verbindung mit dem MTA>
S->M: 220 SMTP Server ESMTP
M->S: EHLO [192.168.1.5]
S->M: 250 – SIZE 35882577
S->M: 250 - AUTH LOGIN PLAIN
S->M: 250 – 8BITMIME
S->M: 250 ENCHANCEDSTATUSCODES
M->S: AUTH LOGIN
S->M: 334 VXNlcm5hbWU6
M->S: bXVzdGVybWFubkBrb21sZS5kZQ==
S->M: 334 UGFzc3dvcmQ6
M->S: lkajsdfvlj
S->M: 235 2.7.0 Authentication successful
M:     <PROXY Zustand>
M->C: 235 2.7.0 Authentication successful
C->M: MAIL FROM:<mustermann@komle.de>
M->S: MAIL FROM:<mustermann@komle.de>
S->M: 250 OK
M->C: 250 OK
C->M: RCPT TO:<musterfrau@komle.de>
M->S: RCPT TO:<musterfrau@komle.de>
S->M: 250 OK
M->C: 250 OK
C->M: DATA
M->S: DATA
S->M: 354 Start mail input; end with <CRLF>.<CRLF>
M->C: 354 Start mail input; end with <CRLF>.<CRLF>
M:     <PROCESS Zustand>
C->M: From: "Max Mustermann" <mustermann@komle.de>
C->M: To: "Erika Musterfrau" <musterfrau@komle.de>
C->M: Subject: Biopsie Ergebnisse für Frau S. Muster
C->M: Date: Mon, 30 Jan 2012 13:14:12 +0100
C->M:
C->M: <Inhalt der KOM-LE Nachricht>
C->M: .
M:     <Die Nachricht wird im Clientmodul aufbereitet>
M->S: X-KOM-LE-Version: 1.0
M->S: MIME-Version: 1.0
M->S: From: "Max Mustermann" <mustermann@komle.de>
M->S: To: "Erika Musterfrau" <musterfrau@komle.de>
M->S: Subject: Verschlüsselte KOM-LE Nachricht
M->S: Date: Mon, 30 Jan 2012 13:14:12 +0100
M->S: Content-Type: application/pkcs7-mime; mime-type=enveloped-
data;name=smime.p7m
M->S: Content-Transfer-Encoding: base64
M->S: Content-Disposition: attachment; filename=smime.p7m
M->S:
M->S: <verschlüsselter Inhalt der KOM-LE Nachricht>
M->S: .
M:     <PROXY Zustand>
S->M: 250 Ok
M->C: 250 Ok
C->M: QUIT
M->S: QUIT
```

```
S->M: 221 Bye
S:     <der MTA schließt die Verbindung mit dem Clientmodul>
M->C: 221 Bye
M:     <das Clientmodul schließt die Verbindung mit dem Clientsystem>
```

Das Senden einer Nachricht wird abgebrochen, weil die Anmeldedaten keine MTA-Adresse erhalten:

```
C:     <das Clientsystem öffnet eine mit TLS geschützte Verbindung mit dem
Clientmodul>
M:     <CONNECT Zustand>
M->C: 220 KOM-LE Clientmodul ESMTP
C->M: EHLO [192.168.1.5]
M->C: 250 – SIZE 35882577
M->C: 250 - AUTH LOGIN PLAIN
M->C: 250 – 8BITMIME
M->C: 250 ENCHANCEDSTATUSCODES
C->M: AUTH LOGIN
M->C: 334 VXNlcm5hbWU6
C->M: bXVzdGVybWFubkBrb21sZS5kZQ==
M->C: 334 UGFzc3dvcmQ6
C->M: lkajsdfvlj
M->C: 501 5.5.4 Benutzername muss die Adresse und die Portnummer des SMTP
Servers
       Enthalten
M:     <das Clientmodul schließt die Verbindung mit dem Clientsystem>
```

Das Senden einer Nachricht wird abgebrochen, weil Verschlüsselungszertifikate weder für mustermann@komle.de noch für musterfrau@komle.de gefunden werden konnten:

```
...
S->M: 354 Start mail input; end with <CRLF>.<CRLF>
M->C: 354 Start mail input; end with <CRLF>.<CRLF>
M:     <PROCESS Zustand>
C->M: From: "Max Mustermann" <mustermann@komle.de>
C->M: To: "Erika Musterfrau" <musterfrau@komle.de>
C->M: Subject: Biopsie Ergebnisse für Frau S. Muster
C->M: Date: Mon, 30 Jan 2012 13:14:12 +0100
C->M:
C->M: <Inhalt der KOM-LE Nachricht>
C->M: .
M:    <Das Clientmodul konnte die Verschlüsselungszertifikate nicht finden>
M->C: 451 Die Nachricht konnte nicht verschlüsselt werden, weil
       Verschlüsselungszertifikate für mustermann@komle.de, musterfrau@komle.de
       nicht zugänglich sind
M->S: RSET
S->M: 250 2.0.0 Flushed
C->M: QUIT
M->S: QUIT
S->M: 221 Bye
S:     <der MTA schließt die Verbindung mit dem Clientmodul>
M->C: 221 Bye
M:     <das Clientmodul schließt die Verbindung mit dem Clientsystem>
```

Das Senden einer Nachricht wird abgebrochen, weil die Verbindung zwischen dem Clientmodul und dem Clientsystem abgebrochen wird:

```
...
M->C: 235 2.7.0 Authentifizierung erfolgreich
C->M: MAIL FROM:<mustermann@komle.de>
M->S: MAIL FROM:<mustermann@komle.de>
S->M: 250 OK
M->C: 250 OK
C->M: RCPT TO:<musterfrau@komle.de>
C:     <das Clientsystem bricht die Verbindung mit dem Clientmodul ab>
```

```
M->S: RCPT TO:<musterfrau@komle.de>
M:     <das Clientmodul schließt die Verbindung mit dem MTA>
```

3.3 Empfangen von Nachrichten

In diesem Kapitel werden Anforderungen an das Clientmodul formuliert, die für den Anwendungsfall „KOM-LE_AF_2 Nachricht empfangen“ [gemSysL_KOMLE] spezifisch sind.

3.3.1 Übersicht

Beim Empfangen von KOM-LE-Nachrichten sorgt das Clientmodul dafür, dass für abgeholte Nachrichten vor der Weiterleitung an das Clientsystem der Vertraulichkeitsschutz aufgehoben und die Integrität geprüft werden. Abbildung 9 stellt die Interaktionen zwischen den am Abholen von KOM-LE-Nachrichten beteiligten Komponenten dar. Aus Sicht des Clientsystems agiert das Clientmodul als POP3-Server, und aus Sicht des POP3-Servers des Fachdienstes (weiter im Text auch als POP3-Server bezeichnet) agiert das Clientmodul als E-Mail-Client. Für Funktionen wie Datentransport, kryptographische Operationen, Kommunikation mit dem Verzeichnisdienst verwendet das Clientmodul entsprechende Dienste der TI-Plattform.

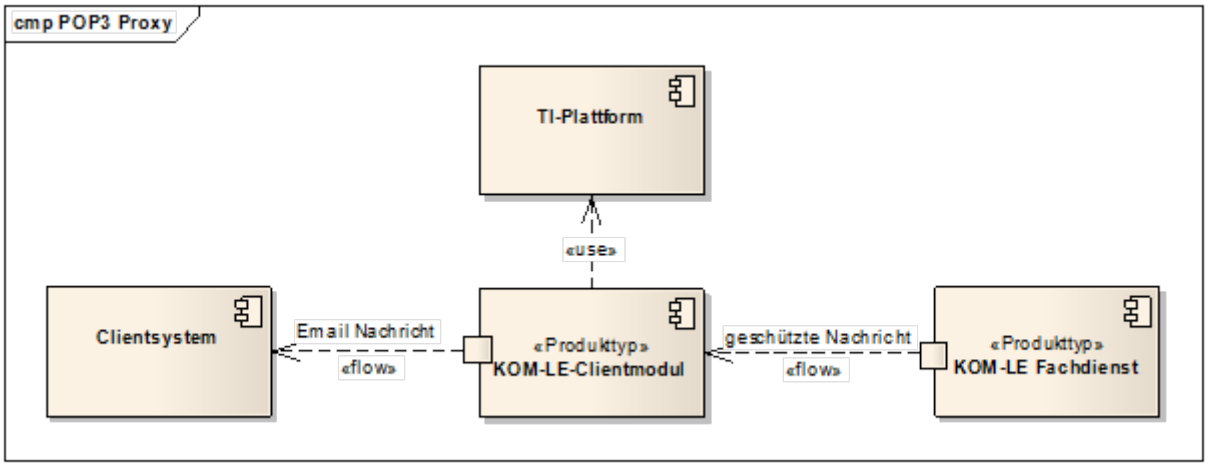

Abbildung 9: Abb_Empfangen_Msg Empfangen von Nachrichten

Beim Abholen von Nachrichten findet die Kommunikation zwischen dem Clientsystem, dem Clientmodul und dem POP3-Server über POP3 statt. Das Clientmodul fungiert als POP3-Proxy, der das Clientsystem mit dem POP3-Server verbindet, die Entschlüsselung und Signaturprüfung für die abgeholten Nachrichten durchführt und die entschlüsselten Nachrichten an das Clientsystem liefert. Die Ergebnisse der Signaturprüfung werden dem Nutzer als Vermerk, der in den Inhalt der Nachricht integriert wird sowie als ein detaillierter Bericht in Form einer angehängten PDF-Datei mitgeteilt.

Dieses Dokument spezifiziert nicht alle Schritte und Einzelheiten der POP3-Kommunikation zwischen dem Clientsystem, dem Clientmodul und dem POP3-Server. Es setzt voraus, dass POP3 und dessen Erweiterungen dem Leser bekannt sind.

Das Clientmodul benachrichtigt den Nutzer über Fehler, die während der Nachrichten-übertragung zwischen dem POP3-Server und dem Clientmodul oder bei der Bearbeitung

der Nachrichten im Clientmodul auftreten. In den meisten Fällen wird das Clientsystem durch POP3-Meldungen über Fehler informiert. Das Clientsystem entscheidet anschließend über das weitere Vorgehen (weitermachen oder abbrechen und den Nutzer über den Fehler informieren).

Beispiel: Verwendet das Clientsystem beim Empfangen von Nachrichten falsche Anmeldungsdaten, bekommt es vom Clientmodul „-ERR Der Nutzer konnte nicht authentifiziert werden“ als Antwort auf sein PASS-Kommando.

Fehler, die bei der Entschlüsselung oder Signaturprüfung einer Nachricht auftreten, werden anders behandelt:

- Kann die Nachricht nicht entschlüsselt werden (z.B. weil der entsprechende HBA nicht zu Verfügung steht), wird durch das Clientmodul eine Fehlernachricht generiert, die die verschlüsselte Nachricht als Anhang enthält. Um die Nachricht nachträglich zu entschlüsseln und ihre Signatur zu prüfen, kann der Nutzer die Nachricht an seine eigene E-Mail-Adresse senden, Maßnahmen treffen damit beim nächsten Abholen der entsprechende Schlüssel gefunden wird und den Abholvorgang wiederholen.
- Wenn die Signaturprüfung der entschlüsselten Nachricht fehlschlägt (z.B. weil die Integrität der Nachricht verletzt wurde, das Signaturzertifikat nicht vorhanden ist, ein OCSP-Responder nicht zur Verfügung steht usw.) wird die entschlüsselte Nachricht dem Clientsystem mit dem entsprechenden Vermerk übergeben.

Das Verhalten des Clientmoduls beim Abholen von Nachrichten kann mit Hilfe der in Abbildung 10 dargestellten Zustandsmuster beschrieben werden. Die im Dokument dargestellten Zustände haben einen illustrativen und nicht normativen Charakter. Die Umsetzung kann sich unterscheiden, solange das Ergebnis das gleiche ist. Die den Zuständen zugeordnete Anforderungen sind normativ, können aber außerhalb des Kontexts dieser Zustände umgesetzt werden.

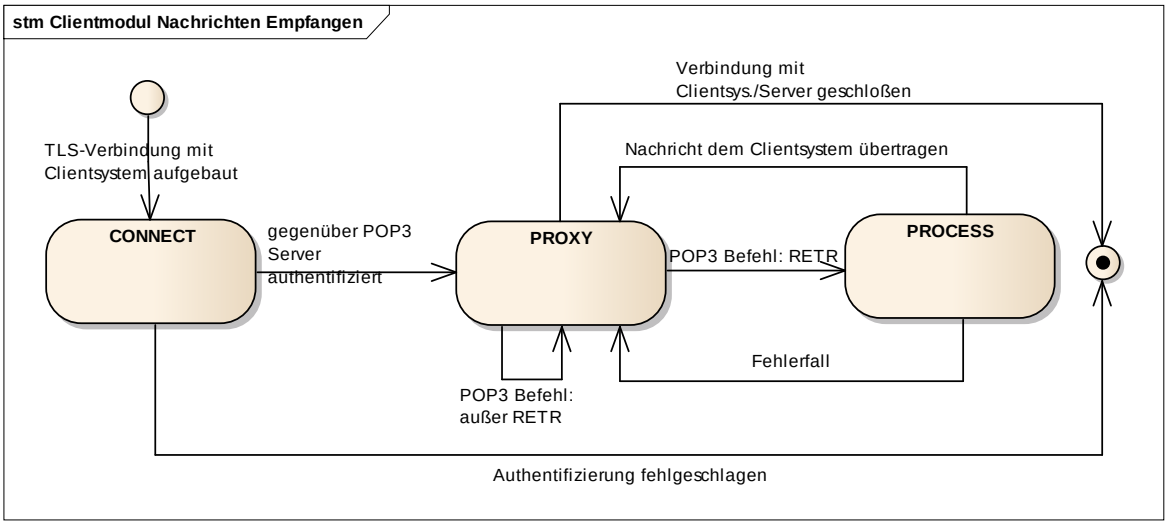

Abbildung 10: Abb_Status_CM_Empfang Zustände Clientmodul beim Nachrichtenempfang

Das Clientmodul lauscht auf einem TCP-Port und wartet bis ein Clientsystem mit ihm eine Verbindung aufbaut. Sobald dies passiert, geht das Clientmodul in den CONNECT-Zustand über und betrachtet die POP3-Verbindung als geöffnet. Die POP3-Verbindung

zwischen dem Clientmodul und dem Clientsystem muss mit TLS erfolgen. Um mit marktüblichen E-Mail-Clients kompatibel zu sein, ist die Unterstützung von TLS 1.0 obligatorisch.

Im CONNECT-Zustand führt das Clientmodul einen POP3-Dialog mit dem Clientsystem, in dem ihm die Anmeldedaten des Nutzers sowie die Adresse und die Portnummer des POP3-Servers mitgeteilt werden. Sobald die Anmeldedaten und die Adresse des POP3-Servers übermittelt sind, baut das Clientmodul eine über TLS geschützte POP3-Verbindung mit dem POP3-Server auf, authentifiziert sich und geht in den PROXY-Zustand über.

Im PROXY-Zustand leitet das Clientmodul POP3-Meldungen und POP3-Antwortcodes zwischen dem Clientsystem und dem POP3-Server hin und her, bis das Clientsystem mit dem RETR-Kommando das Abholen einer Nachricht initiiert. Sobald der POP3-Server beginnt, Inhalte einer Nachricht zu übertragen, geht das Clientmodul in den PROCESS-Zustand über.

Im PROCESS-Zustand wird die Nachricht entschlüsselt, ihre Signatur geprüft und die aufbereitete Nachricht dem Clientsystem übermittelt. Sobald die Nachricht erfolgreich an das Clientsystem übermittelt wurde oder im Fehlerfall, geht das Clientmodul in den PROXY-Zustand zurück.

3.3.2 CONNECT-Zustand

Sobald die TCP-Verbindung zwischen dem Clientsystem und dem Clientmodul aufgebaut wurde, geht das Clientmodul in den CONNECT-Zustand über.

3.3.2.1 Initialisierung

Nachdem die POP3-Verbindung zwischen dem Clientsystem und dem Clientmodul aufgebaut wurde, sendet das Clientmodul dem Clientsystem die POP3-Begrüßung.

Beispiel einer solchen Begrüßung: `+OK KOM-LE Clientmodul POP3`

Das Clientmodul führt einen POP3-Dialog mit dem Clientsystem bis ihm das Clientsystem die Adresse und die Portnummer des POP3-Servers als einen Teil des während des Authentifizierungsverfahrens übertragenen Benutzernamens mitteilt.

Tabelle 3 beschreibt die Antworten, die das Clientmodul dem Clientsystem im CONNECT-Zustand sendet.

Tabelle 3: Tab_POP3_Ant_Init Antworten Clientmodul im CONNECT-Zustand

Clientsystem -> Clientmodul	Clientmodul -> Clientsystem
CAPA	“+OK” Antwortcode mit folgenden CAPA Kennworten: TOP USER SASL PLAIN UIDL

Clientsystem -> Clientmodul	Clientmodul -> Clientsystem
USER, AUTH	Anmeldungsdaten erhalten und Verbindungsaufbau mit dem POP3-Server fortsetzen (siehe Kapitel 3.3.2.2)
QUIT	„+ OK“ Antwortcode senden und die Verbindung mit dem Clientsystem schließen
Andere Meldungen	„-ERR“ Antwortcode

⌦ **KOM-LE-A_2030 POP3-Dialog zur Authentifizierung**

Das Clientmodul MUSS, nachdem die POP3-Verbindung zwischen dem Clientsystem und dem Clientmodul aufgebaut wurde und bis zu dem Punkt an dem das Clientsystem die Bestätigung des Erfolgs oder Misserfolgs seiner Authentifizierung erwartet, einen POP3-Dialog entsprechend Tabelle Tab_POP3_Ant_Init mit dem Clientsystem führen. ⌫

3.3.2.2 Verbindungsaufbau mit dem POP3-Server

Das Clientmodul kann die Verbindung mit dem POP3-Server nur dann aufbauen, wenn ihm das Clientsystem die Adresse des POP3-Servers und die Portnummer des POP3-Dienstes übermittelt. Das Clientmodul erwartet, dass der Domain Name oder die IP-Adresse und die Portnummer während des Authentifizierungsverfahrens als Teil des Benutzernamens übergeben werden.

Das Clientmodul führt das Authentifizierungsverfahren mit dem Clientsystem bis zu dem Punkt, an dem es mit dem entsprechenden Antwortcode die Authentifizierung akzeptieren oder ablehnen muss. Das Clientmodul allein kann das Clientsystem nicht authentifizieren. Die Authentizität der Zugangsdaten kann nur vom POP3-Server überprüft werden. Dazu authentisiert sich das Clientmodul im Auftrag vom Clientsystem gegenüber dem POP3-Server.

Die Server Adresse und die Portnummer des POP3-Dienstes sind als Teil des POP3-Benutzernamens vom Clientsystem zu übergeben. Sie sind vom eigentlichen Benutzer-namen durch das Zeichen ’#’ getrennt und als adresse:port String formatiert.

Um mit SM-B/HBA über den Konnektor kommunizieren zu können, werden dem KOM-LE-Clientmodul ebenfalls als Teil des POP3-Benutzernamens, die

- MandantId
- ClientSystemId
- WorkplaceId
- UserId (optional – ist für einen Zugriff auf HBA erforderlich).

übergeben (siehe Kapitel 3.5 und [gemSpec_Kon] für Details zu MandantId, ClientSystemId, WorkplaceId und UserId). Die Parameter entsprechen denen des aufrufenden Clients und werden voneinander durch das Zeichen ’#’ getrennt. Der Parameter UserId wird nur für den Zugriff auf einen HBA benötigt und kann entfallen wenn kein HBA erforderlich ist (z.B. wenn die Entschlüsselung der empfangenen Nachrichten ausschließlich mit SM-B durchgeführt wird).

Die Reihenfolge der Parameter entspricht dem folgenden Muster:

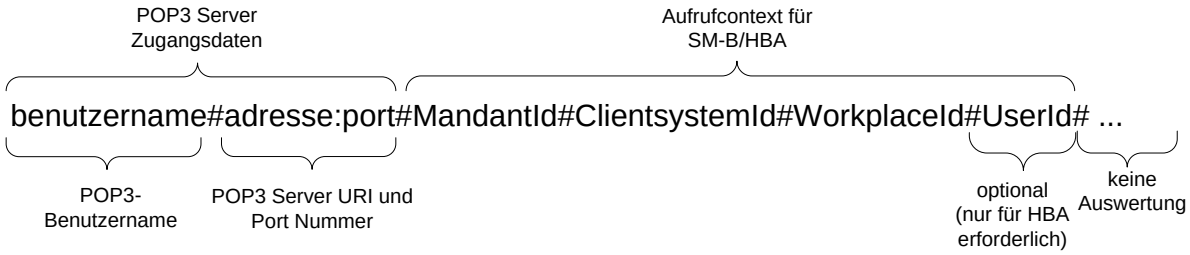

Abbildung 11: Abb_POP3_Nutzer_Name Format des POP3- Benutzernamens

Beispiel:

Bei folgenden Informationen

- Benutzername des Clients = „erik.mustermann@komle.de“,
- Domain Adresse des POP3-Servers = „pop.komle.de“ und Portnummer = 995,
- MandantId = 1,
- ClientsystemId = KOM_LE,
- WorkplaceId = 7,
- UserId = 13

erwartet das Clientmodul, dass das Clientsystem ihm den folgenden POP3-Benutzernamen als String überträgt:

```
erik.mustermann@komle.de#pop.komle.de:995#1#KOM_LE#7#13
```

Enthält der POP3-Benutzername nicht alle erforderlichen Parameter, bricht das KOM-LE-Clientmodul den Empfangsvorgang mit dem -ERR Antwortcode ab. Wenn der erhaltene POP3-Benutzername zusätzliche durch das Zeichen ‚#' abgegrenzte Parameter enthält (z.B. UnknownParameter1#UnknownParamenter2), werden diese Parameter nicht vom Clientmodul ausgewertet und der Empfangsvorgang wird fortgesetzt.

Es gibt mehrere Benutzername/Password-basierte POP3-Authentifizierungsmechanismen:

- Mechanismen, wo die Übertragung von Benutzername und Passwort im Klartext erfolgt (USER/PASS und PLAIN)
- Challenge-Response-Mechanismen, wo der Benutzername im Klartext und das Passwort in Form eines auf vom Server erhaltenen Challenge-basierten Responses übertragen wird (DIGEST-MD5, CRAM-MD5, NTLM).

Die auf Challenge-Response basierten Mechanismen machen das Extrahieren des Passworts aus der Challenge-basierten Response für das Clientmodul unpraktikabel. Deshalb werden für die Clientsystem-Clientmodul-Authentifizierung die PLAIN oder USER/PASS-Mechanismen verwendet.

Sobald das Clientmodul die Anmeldedaten des Nutzers erhält, extrahiert es die Adresse des POP3-Servers und die Portnummer des POP3-Dienstes aus dem Nutzernamen und baut damit die Verbindung zum POP3-Server auf. Die Verbindung wird über TLS geschützt. Details zum Aufbau der TLS-Verbindung werden in Kapitel 4.1.3 beschrieben.

Tabelle 4 enthält POP3-Antwortcodes, die das Clientmodul dem Clientsystem bei einem Verbindungsaufbau mit dem POP3-Server übermittelt.

Tabelle 4: Tab_POP3_Verbindung Antwortcodes für POP3-Server-Verbindungsaufbau

Bedingung	POP3 Antwortcode (Clientmodul -> Clientsystem)
Das Clientsystem hat sich erfolgreich gegenüber dem POP3-Server mit den vom Clientsystem erhaltenen Anmeldungsdaten authentifiziert.	+OK
Das Clientsystem verwendet für die POP3-Authentifizierung einen anderen Mechanismus als USER/PASS oder PLAIN.	-ERR
Die vom Clientsystem erhaltene POP3-Authentifizierungsidentität ist nicht vollständig (POP3 Server Adresse, MandantId, ClientSystemId oder WorkplaceID fehlt – siehe Abbildung 11).	-ERR
Die Verbindung zwischen dem Clientmodul und dem POP3-Server kann nicht aufgebaut werden.	-ERR
Die Authentifizierung gegenüber dem MTA schlägt fehl.	-ERR

Die Verbindungen zwischen dem Clientsystem und dem Clientmodul sowie zwischen dem Clientmodul und dem POP3-Server bleiben solange offen, bis eine der beiden geschlossen oder abgebrochen wird. Sobald eine der beiden Verbindungen geschlossen oder abgebrochen wird, übermittelt das Clientmodul die ausstehenden POP3-Meldungen und schließt die andere Verbindung. Die POP3-Sitzung wird damit für den POP3-Server, das Clientsystem und das Clientmodul beendet.

Beispiel:

Nachdem das Clientmodul das QUIT-Kommando vom Clientsystem erhält und dem POP3-Server übermittelt, bestätigt der POP3-Server das Ankommen des Kommandos mit dem Antwortcode „+OK“ und schließt die Verbindung mit dem Clientmodul. Das Clientmodul übermittelt den Antwortcode „+OK“ an das Clientsystem und schließt die Verbindung mit dem Clientsystem.

⌦ **KOM-LE-A_2031 Unterstützung der Serverteile der Mechanismen USER/PASS und SASL PLAIN**

Das Clientmodul MUSS für die POP3-Authentifizierung des Clientsystems die Serverteile der USER/PASS und SASL-PLAIN-Mechanismen unterstützen. ⌫

⌦ **KOM-LE-A_2032 Extrahieren der Zugangsdaten des POP3-Servers und des Kartenaufrufkontextes**

Das Clientmodul MUSS die Zugangsdaten für den POP3-Server und den Kartenaufrufcontext aus dem vom Clientsystem erhaltenen POP3-Benutzernamen entsprechend Abbildung Abb_POP3_Nutzer_Name extrahieren. ⌫

KOM-LE-A_2033 Verbindungsaufbau mit POP3-Server über Adresse und Portnummer

Das Clientmodul MUSS die POP3-Adresse und die Portnummer, die aus dem vom Clientsystem erhaltenen POP3-Benutzernamen extrahiert wurden (siehe Abbildung Abb_POP3_Nutzer_Name), für die Verbindungsaufbau mit dem POP3-Server verwenden. ⌫

KOM-LE-A_2034 Authentifizierung gegenüber POP3-Server mit Benutzernamen und Passwort

Das Clientmodul MUSS den Benutzernamen, der aus dem vom Clientsystem erhaltenen POP3-Benutzernamen extrahiert wurde (siehe Abbildung Abb_POP3_Nutzer_Name) sowie das vom Clientsystem erhaltene Passwort für die Authentifizierung gegenüber den POP3-Server verwenden. ⌫

KOM-LE-A_2035 Unterstützung der Clientteile der Mechanismen USER/PASS und SASL PLAIN

Das Clientmodul MUSS für das Authentifizierungsverfahren mit dem POP3-Server den Clientteil der USER/PASS und SASL-PLAIN-Mechanismen für POP3-Authentifizierung unterstützen. ⌫

KOM-LE-A_2036 Authentifizierung gegenüber POP3-Server mit anderen Mechanismen als USER/PASS oder SASL PLAIN

Das Clientmodul KANN für das Authentifizierungsverfahren mit dem POP3-Server andere als USER/PASS oder SASL-PLAIN-Authentifizierungsmechanismen benutzen. ⌫

KOM-LE-A_2037 Antwortcodes des Verbindungsaufbaus mit dem POP3-Server

Das Clientmodul MUSS das Clientsystem über das Ergebnis des Verbindungsaufbaus mit dem POP3-Server mit den in der Tabelle Tab_POP3_Verbindung beschriebenen POP3-Antwortcodes informieren. ⌫

KOM-LE-A_2038 Schließen der POP3-Verbindung mit dem Clientsystem

Das Clientmodul MUSS die POP3-Verbindung mit dem Clientsystem aufrechterhalten. Das Schließen der Verbindung ist nur bei folgenden Ausnahmen zulässig:

- Nachdem die Verbindung zwischen dem Clientmodul und dem POP3-Server geschlossen wird. In diesem Fall MUSS das Clientmodul die Verbindung mit dem POP3-Server schließen. Falls es vom POP3-Server erhaltene und dem Clientsystem noch nicht übertragene POP3-Meldungen gibt, MUSS das Clientmodul diese Meldungen dem Clientsystem übertragen, und nur danach die Verbindung mit dem Clientsystem schließen.
- Wenn der POP3-Server innerhalb eines konfigurierbaren Timeouts nicht auf ein POP3-Kommando reagiert. In diesem Fall MUSS das Clientmodul den Antwortcode „- ERR timeout“ an das Clientsystem senden und anschließend die Verbindung schließen.

© gematik - öffentlich

- Wenn die Verbindung zwischen dem Clientmodul und dem POP3-Server noch nicht aufgebaut wurde und das Clientsystem das QUIT-Kommando übermittelt. In diesem Fall MUSS das Clientmodul mit „+OK“ Antwortcode antworten und die Verbindung mit dem Clientsystem schließen. ⌫

⌦ **KOM-LE-A_2039 Schließen der POP3-Verbindung mit dem POP3-Server**

Das Clientmodul MUSS die POP3-Verbindung mit dem POP3-Server aufrechterhalten. Das Schließen der Verbindung ist nur zulässig:

- Nachdem die Verbindung zwischen dem Clientmodul und dem Clientsystem geschlossen wird. In diesem Fall MUSS das Clientmodul die Verbindung mit dem POP3-Server schließen. Falls es vom Clientsystem erhaltene und dem POP3-Server noch nicht übertragene POP3-Kommandos gibt, MUSS das Clientmodul diese Kommandos dem POP3-Server übertragen und nur danach die Verbindung mit dem POP3-Server schließen.
- Wenn das Clientmodul innerhalb eines konfigurierbaren Timeouts keine neuen PoP3-Kommandos sendet. In diesem Fall MUSS das Clientmodul die Verbindung mit dem MTA schließen. ⌫

Nachdem das Clientsystem sich gegenüber dem POP3-Server erfolgreich authentifiziert hat, geht das Clientmodul in den PROXY-Zustand über. Anderenfalls bleibt das Clientmodul im CONNECT-Zustand.

3.3.3 PROXY-Zustand

Im PROXY-Zustand vermittelt das Clientmodul POP3-Meldungen und Antwortcodes zwischen dem Clientsystem und dem POP3-Server. Das Clientmodul bleibt in diesem Zustand bis das Clientsystem das RETR-Kommando sendet und der POP3-Server das Erhalten dieses Kommandos mit dem Antwortcode „+OK“ bestätigt. Das Clientmodul leitet den Antwortcode „+OK“ an das Clientsystem weiter und geht in den PROCESS-Zustand über.

In diesem Zustand kann das Clientmodul vom Clientsystem das TOP-Kommando erhalten, das `<MsgID>` und `<N>` als Parameter hat. Es fordert den POP3-Server zur Übertragung des Headers und von `<N>` Nachrichtenzeilen der durch `<MsgID>` identifizierten Nachricht auf. Um sicherzustellen, dass das Clientmodul keine Teile einer verschlüsselten S/MIME-Nachricht bekommt, wird der Parameter `<N>` vom Clientmodul immer auf 0 gesetzt.

⌦ **KOM-LE-A_2040 Übermittlung von POP3-Kommandos und -Meldungen nach erfolgreicher Authentifizierung**

Das Clientmodul MUSS, nachdem das Authentifizierungsverfahren mit dem Clientsystem erfolgreich beendet ist, alle vom Clientsystem erhaltenen POP3-Kommandos, mit Ausnahme des TOP-Kommandos, bzw. alle vom POP3-Server erhaltenen POP3-Meldungen, mit Ausnahme von Inhalten vom E-Mail-Nachrichten, ohne jegliche Veränderungen dem POP3-Server bzw. dem Clientsystem übermitteln. ⌫

⌦ **KOM-LE-A_2041 Setzen des Parameters <N> des TOP-Kommandos auf Null**

Das Clientmodul MUSS, wenn es vom Clientsystem ein TOP <MsgID> <N> Kommando mit einem von Null abweichenden Parameter <N> erhält, den Wert des Parameters <N> auf Null setzen, bevor das Kommando dem POP3-Server übermittelt wird. ⌫

Hinweis für Implementierung
Wegen eines Thunderbird bugs:
Das getrennte Laden von Header und Body ist in Thunderbird nicht korrekt implementiert. Möglicher Bugfix im CM: Bei TOP 0 den Msg Header ändern: MIME Element(`MIME-Version: 1.0`) aus Header entfernen, dann klappt das nachladen.

3.3.4 PROCESS-Zustand

Im PROZESS-Zustand nimmt das Clientmodul die Inhalte der vom POP3-Server abgerufenen Nachricht entgegen, entschlüsselt die Nachricht, prüft deren Integrität, fügt einen Vermerk sowie einen PDF-Anhang mit dem Ergebnis der Signaturprüfung in die Nachricht ein und leitet die aufbereitete Nachricht dem Clientsystem weiter. Im Erfolgsfall wird das Clientsystem über das erfolgreiche Abholen der Nachricht informiert. Im Fehlerfall wird das Clientsystem mit dem entsprechenden Antwortcode über den Fehler informiert.

3.3.4.1 Empfang und Weiterleitung einer Nachricht

Nachdem der POP3-Server das Erhalten des RETR-Kommandos mit dem Antwortcode „+OK“ bestätigt, erwartet das Clientmodul, dass der POP3-Server mit der Übertragung der Nachricht beginnt. Die Inhalte der Nachricht werden im Clientmodul zwischengespeichert. Wenn die Nachricht eine entsprechend dem KOM-LE-S/MIME-Profil geschützte Nachricht ist, bereitet das Clientmodul die erhaltene Nachricht auf und übermittelt sie anschließend dem Clientsystem. Wenn es keine KOM-LE-S/MIME-Nachricht ist, wird sie ohne jegliche Änderungen dem Clientsystem übermittelt.

Nachdem die Nachricht dem Clientsystem übermittelt wurde, löscht das Clientmodul die zwischengespeicherten Nachrichtinhalte und geht in den PROXY-Zustand zurück.

3.3.4.2 Aufbereitung einer Nachricht

Das Clientmodul soll zwischen den KOM-LE S/MIME und anderen Nachrichten unterscheiden. Wenn die angekommene Nachricht eine KOM-LE-S/MIME-Nachricht ist, entschlüsselt das Clientmodul ihre Inhalte und führt die Prüfung ihrer Signatur durch. Die KOM-LE-S/MIME-Nachrichten sind anhand des `X-KOM-LE-Version` Header-Elements erkennbar. Wenn die ankommende Nachricht keine KOM-LE-S/MIME-Nachricht ist, soll sie ohne weitere Veränderungen dem Clientsystem übermittelt werden.

Für die Entschlüsselung und die Signaturprüfung verwendet das Clientmodul die Dienste der TI-Plattform, die dem Clientmodul über Schnittstellen des Konnektors zur Verfügung gestellt werden.

3.3.4.2.1 Entschlüsselung

Für die Entschlüsselung der ankommenden Nachricht wird der private Schlüssel PrK.HCI.ENC bzw. C.HP.ENC verwendet, der dem Verschlüsselungszertifikat der Institution bzw. des Leistungserbringers zugeordnet ist. Der Zugriff auf die entsprechende Karte und die Entschlüsselung erfolgen über die Aufrufe der entsprechenden Operationen der Außenschnittstelle des Konnektors. Eine detaillierte Beschreibung erfolgt im Kapitel 3.5.4.

Wenn die Nachricht für mehrere Empfänger verschlüsselt wurde, liegt es in der Verantwortung des Clientmoduls sicherzustellen, dass die Nachricht mit dem Schlüssel des den Abholvorgang auslösenden Nutzers entschlüsselt wird. Der erforderliche Schlüssel kann mit Hilfe des im KOM-LE-S/MIME-Profil beschriebenen `recipient-emails` Attributs im `EnvelopedData` CMS-Objekt identifiziert werden. Das `EnvelopedData` CMS-Objekt enthält die verschlüsselten Inhalte und im `recipient-emails` Attribut werden die Zusammenhänge zwischen den E-Mail-Adressen der Empfänger und den verwendeten Verschlüsselungszertifikaten definiert. Das ermöglicht die Identifizierung des erforderlichen Verschlüsselungszertifikats, dessen zugehöriger privater Schlüssel für die Entschlüsselung verwendet werden soll. Dadurch kann vermieden werden, dass die Nachricht mit dem freigeschalteten Schlüssel eines Empfängers entschlüsselt wird, der nicht derjenige ist, der den Abholvorgang ausgelöst hat. Das Clientmodul geht davon aus, dass der Nutzername, der für die POP3-Authentifizierung verwendet wurde, der E-Mail-Adresse des Empfängers entspricht und benutzt ihn, um den entsprechenden `RecipientIdentifier` aus dem `recipient-emails` Attribut auszulesen. Wenn es keinen `RecipientIdentifier` gibt, der dem POP3-Nutzernamen des Empfängers entspricht, wird die Entschlüsselung als fehlgeschlagen betrachtet.

Wenn die Entschlüsselung fehlschlägt, wird dem Clientsystem die verschlüsselte Nachricht im Anhang einer Fehlernachricht übermittelt. Hierzu wird die angekommene KOM-LE-S/MIME-Nachricht als eine `message/rfc822` MIME-Einheit in eine `multipart/mixed` MIME-Nachricht verpackt, die zusätzlich eine `text/plain` MIME-Einheit mit der Fehlermeldung enthält. Die `orig-date`, `from`, `sender`, `reply-to`, `to` und `cc` Header-Elemente der neuen Nachricht werden aus der ursprünglichen Nachricht übernommen. Der Betreff der neuen Nachricht enthält die Zeichenkette „Die Nachricht konnte nicht entschlüsselt werden“.

<u>Beispiel</u>:

Kann eine Nachricht auf Grund des fehlenden HBA mit dem erforderlichen privaten Schlüssel nicht im Clientmodul entschlüsselt werden, wird die Nachricht wie folgt dem Clientsystem übermittelt:

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="unique-boundary-1"
Subject: Die Nachricht konnte nicht entschlüsselt werden
Date: Fri, 9 Feb 2012 12:07:17 +0100
From: mustermann@komle.de
To: musterfrau@komle.de

This is a multi-part message in MIME format.

--unique-boundary-1
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
```

```
Der f=FCr die Entschl=FCsselung der Nachricht ben=F6tigte Schl=FCssel =
wurde nicht gefunden. =DCberpr=FCfen Sie ob die entsprechende Karte =
gesteckt ist und leiten Sie diese Nachricht an Ihre eigene Email Adresse =
(musterfrau@komle.de) weiter. Beim n=E4chsten Abholen wird der =
Verschl=FCsselungsvorgang wiederholt.

--unique-boundary-1
Content-Type: message/rfc822

X-KOM-LE-Version: 1.0
MIME-Version: 1.0
Content-Type: application/pkcs7-mime; name="smime.p7m"; name="smime.p7m"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7m"
Subject: Verschlüsselte KOM-LE Nachricht
Date: Fri, 9 Feb 2012 12:07:17 +0100
From: mustermann@komle.de
To: musterfrau@komle.de

567GhIGfHfYT6ghyHhHUujpfyF4f8HHGTrfvhJhjH776tbB9HG4VQbnj7
77n8HHGT9HG4VQpfyF467GhIGfHfYT6rfvbnj756tbBghyHhHUujhJhjH
HUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H7n8HHGghyHh
...
9efmAAAAAAAAAAAAAA==

--unique-boundary-1
```

⌦ KOM-LE-A_2042 Entschlüsselung einer KOM-LE-SMIME-Nachricht

Das Clientmodul MUSS eine vom POP3-Server erhaltene und dem KOM-LE-S/MIME-Profil entsprechende E-Mail entschlüsseln. Nachrichten, die nicht dem KOM-LE-S/MIME-Profil entsprechen, sind ohne Veränderung an das Clientsystem weiterzuleiten. ⌫

⌦ KOM-LE-A_2043 Beachtung des recipient-emails Attributs bei der Entschlüsselung

Das Clientmodul MUSS bei der Entschlüsselung das recipient-emails Attribut des EnvelopedData-CMS-Objekts beachten, um die Nachricht mit dem Schlüssel des Nutzers, der den Abholvorgang ausgelöst hat, zu entschlüsseln. ⌫

⌦ KOM-LE-A_2044 E-Mail-Adresse des den Abholvorgang auslösenden Nutzers

Das Clientmodul MUSS den vom Clientsystem erhaltenen POP3-Usernamen (ohne den #server:port#... Teil) als die E-Mail-Adresse des den Abholvorgang auslösenden Nutzers betrachten. ⌫

⌦ KOM-LE-A_2045 Entschlüsselung nur mit Schlüsseln des abholenden Nutzers

Das Clientmodul DARF für die Entschlüsselung einer Nachricht Schlüssel NICHT verwenden, wenn sie von anderen Nutzern stammen als von dem der den Abholvorgang ausgelöst hat. ⌫

⌦ KOM-LE-A_2179 Vermerk in der Nachricht bei erfolgreicher Entschlüsselung

Das Clientmodul MUSS bei erfolgreicher Entschlüsselung der KOM-LE-Nachricht den Vermerk „Die Nachricht wurde entschlüsselt.“ an den Text der Nachricht anhängen.⌫

⌦ **KOM-LE-A_2046 Aufbau der Fehlernachricht bei fehlgeschlagener Entschlüsselung**

Das Clientmodul MUSS eine empfangene, dem KOM-LE-S/MIME-Profil entsprechende Nachricht, die z.B. auf Grund des fehlenden Schlüssels nicht entschlüsselt werden kann, als eine message/rfc822 MIME-Einheit in einer neuen multipart/mixed MIME-Nachricht dem Clientsystem übermitteln. Zusätzlich muss diese neue multipart/mixed MIME-Nachricht eine text/plain MIME-Einheit mit dem Fehlertext enthalten. Die orig-date, from, sender, reply-to, to und cc Header-Elemente der neuen multipart/mixed Nachricht werden aus der empfangenen Nachricht übernommen. Das subject Header-Element der neuen multipart/mixed Nachricht erhält den Wert „Die Nachricht konnte nicht entschlüsselt werden“. ⌫

Durch das Versenden einer solchen Fehlernachricht erhält der Nutzer die Möglichkeit, die E-Mail entweder vom Server zu löschen oder durch das Senden an die eigene E-Mail-Adresse und das anschließende Abholen die Aufbereitung zu wiederholen. Ein anderer Weg wäre die Nachrichten, die nicht vom Clientmodul aufbereitet werden konnten, auf dem Mail Server zu belassen und beim nächsten Abholen die Aufbereitung zu wiederholen. Der Nachteil eines solchen Ansatzes wäre, dass unter Umständen „E-Mail-Leichen“ entstehen. Hierbei handelt es sich um E-Mails, die z.B. auf Grund des Verlustes des erforderlichen HBA nicht mehr aufbereitet werden können und deswegen auf dem E-Mail-Server verbleiben würden.

Tabelle 5 enthält die Fehlertexte, die in die Nachricht eingeführt werden, wenn die Entschlüsselung nicht durchgeführt werden konnte.

Tabelle 5: Tab_Fehlertext_Entschl Fehlertexte für Entschlüsselungsfehler

Bedingung	Fehlertexte
Die KOM-LE-Nachricht konnte auf Grund eines nicht verfügbaren Schlüssels nicht entschlüsselt werden.	Der für die Entschlüsselung der Nachricht benötigte Schlüssel wurde nicht gefunden. Überprüfen Sie ob die entsprechende Karte gesteckt ist und leiten Sie diese Nachricht an Ihre eigene E-Mail-Adresse (<Email Adresse>) weiter. Beim nächsten Abholen wird der Verschlüsselungsvorgang wiederholt.
Die KOM-LE-Nachricht konnte aufgrund des falschen Formats nicht entschlüsselt werden (z.B. enthält die Nachricht das X-KOM-LE-Version Header-Element, entspricht aber nicht dem KOM-LE-S/MIME-Profil).	Die Nachricht wurde als eine verschlüsselte KOM-LE-Nachricht gekennzeichnet, konnte aber auf Grund des falschen Formats nicht entschlüsselt werden. Die Verschlüsselte Nachricht befindet sich im Anhang.
Der Konnektor steht für die Entschlüsselung nicht zur Verfügung.	Die Entschlüsselung konnte nicht erfolgen, weil der Konnektor nicht antwortet. Stellen Sie sicher, dass der Konnektor wieder zur Verfügung steht und leiten Sie diese Nachricht an Ihre eigene E-Mail-Adresse (<Email Adresse>) weiter. Beim nächsten Abholen wird der Verschlüsselungsvorgang wiederholt.

⌦ **KOM-LE-A_2047 Fehlertexte bei fehlgeschlagener Entschlüsselung**

Das Clientmodul MUSS bei fehlgeschlagener Entschlüsselung entsprechend der jeweiligen Bedingung die in Tabelle Tab_Fehlertext_Entschl definierten Fehlertexte

in die `text/plain` MIME-Einheit der multipart/mixed MIME-Fehlernachricht aufnehmen. ⌫

3.3.4.2.2 Integritätsprüfung

Nachdem die angekommene Nachricht erfolgreich entschlüsselt wurde, prüft das Clientmodul ihre Integrität. Dabei werden die digitale Signatur der Nachricht, der Zertifizierungspfad für das Signaturzertifikat und die Integrität des `recipient-emails` Attributs geprüft. Für die Signaturprüfung der Nachricht wird das im CMS-Objekt mitgelieferte C.HCI.OSIG-Institutionszertifikat benutzt. Die Prüfung der Signatur erfolgt über die Aufrufe der entsprechenden Operationen der Außenschnittstelle des Konnektors. Eine detaillierte Beschreibung erfolgt Kapitel 3.5.2.

Das Ergebnis der Signaturprüfung und des Abgleichs des `recipient-emails` Attributs wird als Vermerk, der den Text der Nachricht ergänzt, dem Empfänger mitgeteilt. Zusätzlich wird eine PDF-Datei mit einem detaillierten Signaturprüfungsbericht als Anhang in die Nachricht eingefügt.

Der Dateiname des Signaturprüfungsberichtes ist Signaturpruefungsbericht.pdf und hat die folgende Struktur:

Tabelle 6: Tab_Strukt_Sig_Prüf_Report Struktur Signaturprüfbericht

Gesamtergebnis Abhängig vom Ergebnis der Signaturprülfung ist hier der Text entsprechend Vermerk aus Tabelle Tab_Verm_Sig_Prüf Vermerke mit Ergebnissen der Signaturprüfung einzufügen	
A. Signaturdeteils	
Signaturzeitpunkt laut Unterzeichner:	Datum (tt.mm.jj) Uhrzeit (hh:mm:ss)
Datum der Signaturprüfung:	Datum (tt.mm.jj) Uhrzeit (hh:mm:ss)
Dokumentgröße in Bytes:	z.B.: 1987
Hashalgorithmus:	z.B.: SHA-256
Signaturalgorithmus:	z.B.: RSA Verschlüsselung mit SHA-256 Hash
Schlüssellänge in Bits:	z.B.: 2048
	Ergebnis der Prüfung der mathematischen Prüfung der Signatur (z.B.: Der vom Unterzeichner signierte Hashwert passt zu den signierten Daten)
B. Zertifikatsdetails	
Signaturzertifikatsdetails	
Inhaber des Zertifikats:	cn aus Zertifikat (z.B.: cn=Egon Mustermann)
Typ:	Nutzerzertifikat
Seriennummer (hex):	z.B.: 0x1597f
Zertifikat frühenstens gülltig seit:	Datum (tt.mm.jj) Uhrzeit (hh:mm:ss)
Zertifikat längstens gülltig bis:	Datum (tt.mm.jj) Uhrzeit (hh:mm:ss)
Zeitpunkt der Gültigkeitsprüfung:	Datum (tt.mm.jj) Uhrzeit (hh:mm:ss)

Aussteller des Zertifikats:	dn des Ausstellers (z.B: cn=gematik SMC-B CA, o=gematik, c=de)
	Ergebnis der zeitlichen Gültigkeitsprüfung (z.B.: Zertifikat zeitlich gültig)
	Ergebnis der Prüfung der Signatur des Ausstellerzertifikats (z.B.: Das Zertifikat hat eine gültige Signatur vom Austellerzertifikat)
Herausgeberzertifikatsdetails (für alle Zertifikate in der Kette)	
Inhaber des Zertifikats:	cn aus Zertifikat (z.B.: cn=Egon Mustermann)
Typ:	Austellerzertifikat
Seriennummer (hex):	z.B.: 0x25d97f
Zertifikat frühenstens gülltig seit:	Datum (tt.mm.jj) Uhrzeit (hh:mm:ss)
Zertifikat längstens gülltig bis:	Datum (tt.mm.jj) Uhrzeit (hh:mm:ss)
Zeitpunkt der Gültigkeitsprüfung:	Datum (tt.mm.jj) Uhrzeit (hh:mm:ss)
Aussteller des Zertifikats:	dn des Ausstellers
	Ergebnis der zeitlichen Gültigkeitsprüfung (z.B.: Zertifikat zeitlich gültig)
	Ergebnis der Prüfung der Signatur des Ausstellerzertifikats (z.B.: Das Zertifikat hat eine gültige Signatur vom Austellerzertifikat)
C. Online-Sperrabfrage für Signaturzertifikat	
Zugriff erfolgte am:	Datum (tt.mm.jj) Uhrzeit (hh:mm:ss)
OCSP-Status des Zertifikats:	good|revoked|unknown
Dienst:	URL OCSP-Responder (z.B.: http://www.gematik-smcb-ocsp.de)

Falls der Zertifikatstatus des Signaturzertifikates nicht geprüft werden kann (z.B. der OCSP-Responder ist unerreichbar), die mathematische Prüfung der Signatur aber erfolgreich durchgeführt wurde, wird ein entsprechender Vermerk in der Body der Nachricht eingetragen.

Tabelle 7 stellt die Vermerke entsprechend den Ergebnissen der Signaturprüfung dar.

Tabelle 7: Tab_Verm_Sig_Prüf Vermerke mit Ergebnissen der Signaturprüfung

Ergebnis	Vermerk
Die Signatur der Nachricht wurde erfolgreich geprüft.	Die Signatur wurde erfolgreich geprüft.
Die Integrität der Nachricht wurde verletzt.	Die Prüfung der Signatur hat ergeben, dass die Nachricht manipuliert wurde.
Die digitale Signatur ist nicht vorhanden.	Die Nachricht ist nicht signiert. Die Nachricht ist deshalb eventuell manipuliert worden.
Die digitale Signatur konnte aufgrund des falschen Formats nicht geprüft werden.	Die Signatur der Nachricht konnte aufgrund eines falschen Formats nicht geprüft werden. Die Nachricht ist deshalb eventuell manipuliert

Ergebnis	Vermerk
	worden.
Der Zertifizierungspfad des Signaturzertifikats kann nicht validiert werden (abgelaufenes Zertifikat, der Zertifizierungspfad konnte nicht aufgebaut werden usw.).	Die Signatur der Nachricht wurde geprüft. Die dabei durchgeführte Integritätsprüfung der Nachricht war erfolgreich. Der Status des zur Signatur verwendeten Zertifikats konnte nicht vollständig durchgeführt werden, weil nicht alle am Signaturprozess beteiligten Zertifikate validiert werden konnten.
Die digitale Signatur ist mathematisch korrekt, der Zertifikatsstatus des Signaturzertifikats konnte aber nicht geprüft werden.	Die Signatur der Nachricht wurde geprüft. Die dabei durchgeführte Integritätsprüfung der Nachricht war erfolgreich. Der Status des zur Signatur verwendeten Zertifikats konnte nicht vollständig geprüft werden, weil zum Prüfungszeitpunkt nicht alle erforderlichen technischen Ressourcen verfügbar waren.
Die digitale Signatur ist mathematisch korrekt und der Zertifikatsstatus des Signaturzertifikats konnte erfolgreich geprüft werden, aber beim Vergleich der Header-Elemente orig-date, from, sender, reply-to, to und cc der äußeren Nachricht mit denen der inneren Nachricht wurden Abweichungen festgestellt.	Die Signatur der Nachricht wurde geprüft. Die Prüfung hat ergeben, dass die Nachricht nach dem Verschlüsseln manipuliert wurde. Möglicherweise wurde die verschlüsselte Nachricht auch an einen nicht empfangs-berechtigten Personenkreis versendet.
Die digitale Signatur ist mathematisch korrekt und der Zertifikatsstatus des Signaturzertifikats konnte erfolgreich geprüft werden, aber das recipient-emails Attribut aus signerInfos enthält nicht die gleichen Werte wie das recipient-emails Attribut aus dem enveloped-data CMS-Objekt.	Die Signatur der Nachricht wurde geprüft. Die Prüfung hat ergeben, dass die Nachricht manipuliert wurde, um einem anderen Nutzer das Entschlüsseln der Nachricht mit einem Schlüssel, der nicht in seiner Besitz ist, zu ermöglichen.

Es folgt ein Beispiel einer entschlüsselten `multipart/mixed` Nachricht deren Signatur erfolgreich geprüft wurde. Die Nachricht enthält eine `text/plain` Einheit im Nachrichtentext, einen Arztbrief als PDF-Anhang sowie den Signaturprüfungsbericht ebenfalls als PDF-Anhang.

```
Date: Fri, 9 Feb 2012 12:07:17 +0100
MIME-Version: 1.0
From: mustermann@komle.de
To: musterfrau@komle.de
Subject: Arztbrief H. Muster
Content-Type: multipart/mixed;
boundary="unique-boundary-1"

This is a multi-part message in MIME format.
--unique-boundary-1
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

Sehr Geehrte Frau Dr. Musterfrau,

hiermit sende ich Ihnen den Arztbrief f=FCr Herrn H. Muster.

Mit Freundlichen Gr=FC=DFen
Dr. med. Mustermann
```

```
Arzt f=FCr Allgemeinmedizin

-----------------------------------------------
Die Nachricht wurde entschl=FCsselt
Die Signatur wurde erfolgreich gepr=FCft.
--unique-boundary-1
Content-Type: application/pdf;
name="Arztbrief_Muster.pdf"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
filename="Arztbrief_Muster.pdf"

JVBERi0xLjQNCiXDpMO8w7bDnw0KMiAwIG9iag0KPDwgL0xlbmd0aCAzIDAgUg0KICAgL0Zp
bHRlciAvRmxhdGVEZWNvZGUNCj4+DQpzdHJlYW0NCnicrVhda1sxDH0P5D/4uQ+3lvxxfaEM
...
OEJCQUExQzY0NDU+IF0NCj4+DQpzdGFydHhyZWYNCjIyNDU3Mg0KJSVFT0YNCg==
--unique-boundary-1
Content-Type: application/pdf;
name="Signaturpruefungsbericht.pdf"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
filename="Signaturpruefungsbericht.pdf"

CjwhLS0gc2F2ZWQgZnJvbSB1cmw9KDAwMzgpaHR0cDovL2l3aS53aXdpLmh1LWJlcmxpbi5kZS9+
ZXZkb2tpbS8gLS0+CjxodG1sPjxoZWFkPjxtZXRhIGh0dHAtZXF1aXY9IkNvbnRlbnQtVHlwZSIg
...
PC9saT4KPC91bD4KCgo8L2JvZHk+PC9odG1sPg==
--unique-boundary-1
```


⌦ KOM-LE-A_2048 Prüfung der Signatur einer KOM-LE-Nachricht

Das Clientmodul MUSS die Integrität der KOM-LE-Nachricht prüfen. Dabei müssen die digitale Signatur selbst, der Zertifizierungspfad für das verwendete Signaturzertifikat, die Integrität des Headers der äußeren Nachricht und die Integrität des recipient-emails Attributs geprüft werden.

Bei der Prüfung der Integrität des Headers der äußeren Nachricht sind die Header-Elemente orig-date, from, sender, reply-to, to und cc mit denen der signierten inneren Nachricht zu vergleichen.

Bei der Prüfung der Integrität des recipient-emails Attributs sind die Werte dieses Attributs aus signerInfos und aus dem enveloped-data CMS-Objekt miteinander zu vergleichen. ⌫

⌦ KOM-LE-A_2049 Ergebnis der Signaturprüfung einer KOM-LE-Nachricht

Das Clientmodul MUSS das Ergebnis der Signaturprüfung der KOM-LE-Nachricht als Vermerk an den Text der Nachricht anhängen. Zusätzlich MUSS das Clientmodul eine PDF-Datei mit einem detaillierten Signaturprüfungsbericht als Anhang mit dem Namen Signaturpruefungsbericht.pdf in die Nachricht einfügen. ⌫

⌦ KOM-LE-A_2180 Struktur des Signaturprüfberichts

Der vom Clientmodul in einer PDF-Datei zu erzeugende Signaturprüfungsbericht MUSS der in Tabelle Tab_Strukt_Sig_Prüf_Report Struktur Signaturprüfbericht beschriebenen Struktur entsprechen. ⌫

⌦ **KOM-LE-A_2050 Vermerke des Ergebnisses der Signaturprüfung einer KOM-LE-Nachricht**

Das Clientmodul MUSS abhängig vom Ergebnis der Signaturprüfung einer KOM-LE-Nachricht die in Tabelle Tab_Verm_Sig_Prüf definierten Vermerke an den Nachrichtentext der KOM-LE-Nachricht anfügen. ⌫

3.3.5 Beispiele

Das Clientsystem (C) verbindet sich mit dem Clientmodul (M) und holt vom POP3-Server (S) eine Nachricht (im Beispiel werden auch die Zustände des Clientmoduls dargestellt):

```
C:     <das Clientsystem öffnet eine mit TLS geschützte Verbindung mit dem
       Clientmodul>
M:     <CONNECT Zustand>
M->C: +OK KOM-LE Clientmodul POP3
C->M: CAPA
M->C: +OK Capability list follows
M->C: TOP
M->C: USER
M->C: SASL PLAIN
M->C: UIDL
M->C: .
C->M: USER mustermann@komle.de#pop.komle.de:110#1#KOM-LE#7
M->C: +OK
C->M: PASS password
M:     <das Clientmodul öffnet eine mit TLS geschützte Verbindung mit dem POP3
       Server>
S->M: +OK POP Server Ready
M->S: CAPA
S->M: +OK Capability list follows
S->M: TOP
S->M: USER
S->M: SASL PLAIN CRAM-MD5
S->M: UIDL
S->M: RESP-CODES
S->M: .
M->S: USER mustermann@komle.de
S->M: +OK
M->S: PASS password
S->M: +OK Maildrop ready
M:     <PROXY Zustand>
M->C: +OK Maildrop ready
C->M: STAT
M->S: STAT
S->M: +OK 1 13950
M->C: +OK 1 13950
C->M: LIST
M->S: LIST
S->M: +OK
M->C: +OK
S->M: 1 13950
M->C: 1 13950
S->M: .
M->C: .
C->M: UIDL
M->S: UIDL
S->M: +OK
M->C: +OK
S->M: 1 01SDF8-1RiSd50vfv-00FGJN
M->C: 1 01SDF8-1RiSd50vfv-00FGJN
S->M: .
```

```
M->C: .
C->M: RETR 1
M->S: RETR 1
S->M: +OK
M->C: +OK
M:     <PROCESS Zustand>
S->M: <Inhalt der verschlüsselten KOM-LE Nachricht>
S->M: .
M:     <die Nachricht wird im Clientmodul aufbereitet>
M->C: <Inhalt der KOM-LE Nachricht>
M->C: .
M:     <PROXY Zustand>
C->M: QUIT
M->S: QUIT
S->M: +OK
S:     <der POP3 Server schließt die Verbindung mit dem Clientmodul>
M->S: +OK
M:     <das Clientmodul schließt die Verbindung mit dem Clientsystem>
```

Während des Löschens einer Nachricht wird die Verbindung zwischen dem Clientmodul und dem POP3-Server abgebrochen:

```
...
C->M: UIDL
M->S: UIDL
S->M: +OK
M->C: +OK
S->M: 1 01SDF8-1RiSd50vfv-00FGJN
M->C: 1 01SDF8-1RiSd50vfv-00FGJN
S->M: .
M->C: .
C->M: DELE 1
C:     <die Verbindung zwischen dem Clientmodul und dem Clientsystem wird
       abgebrochen>
M->S: DELE 1
M:     <die Verbindung zwischen dem Clientmodul und dem POP3 Server wird
       geschlossen>

.
```

3.4 Übermittlung von Kontaktdaten

Ein KOM-LE-Nutzer soll die Möglichkeit haben in seinem Clientsystem die Suche nach den E-Mail-Adressen der Empfänger seiner KOM-LE-Nachrichten durchzuführen. Die TI-Plattform stellt einen Verzeichnisdienst zu Verfügung, der unter anderem Einträge mit Kontaktdaten von KOM-LE-Nutzern enthält. Der Verzeichnisdienst kann über LDAP abgefragt werden und kann somit als Adressbuch für KOM-LE benutzt werden. Eine detaillierte Beschreibung des Verzeichnisdienstes der TI-Plattform befindet sich in [gemSpec_VZD]. Um LDAP-Anfragen gegenüber dem Verzeichnisdienst durchzuführen, fungiert der Konnektor als LDAP-Proxy wie in [gemSpec_Kon] beschrieben.

Der Verzeichnisdienst kann direkt von Clientsystemen, die die entsprechenden LDAP-Suchanfragen generieren, angefragt werden. Das LDAP-Schema des Verzeichnisdienstes wird in [gemSpec_VZD] beschrieben.

3.5 Kryptographischen Schnittstellen des Konnektors

Das digitale Signieren und die Verschlüsselung von Nachrichten sowie deren Entschlüsselung und die Prüfung ihrer digitalen Signaturen beinhalten den Zugriff auf die SOAP-Schnittstellen des Konnektors, die die folgenden Operationen zu Verfügung stellen:

- `SignDocument` - Erzeugung einer digitalen Signatur,
- `VerifyDocument` – Prüfung einer digitalen Signatur,
- `EncryptDocument` – Verschlüsselung und
- `DecryptDocument` - Entschlüsselung.

Die Verschlüsselung und das digitale Signieren erfordern dabei den Zugriff auf eine SM-B und/oder einen HBA mit dem erforderlichen Schlüsselmaterial. Zur Erstellung einer digitalen Signatur ist der Zugriff auf den geheimen Schlüssel PrK.HCI.OSIG einer SM-B erforderlich. Für die Verschlüsselung ist der Zugriff auf den geheimen Schlüssel Prk.HCI.ENC einer SM-B oder Prk.HP.ENC eines HBA notwendig.

Der Zugriff auf den entsprechenden geheimen Schlüssel erfolgt während der Durchführung der `SignDocument` und `DecryptDocument` Operationen. Die Eingangsparameter der beiden Operationen beinhalten das `Context` Element (Aufrufkontext). Der Aufrufkontext umfasst die Angaben zu Mandanten (`MandantId`), Arbeitsplatz (`WorkplaceId`), Anwendung (`ClientSystemId`) und Identifikation des Benutzers (`UserId`). Die Angaben zur Identifikation des Benutzers (`UserId`) sind optional und nur für Aufrufe, die einen Zugriff auf den HBA brauchen, erforderlich. Die Elemente des Aufrufkontexts werden dem Clientmodul als Teile des MTA- bzw. POP3-Benutzernamens übertragen (siehe Kapitel 3.2.2.2, 3.3.2.2).

Zur Identifikation der Karte benötigen die Operationen zusätzlich den Paramemter `cardHandle`. Das `cardHandle` gilt für die Dauer des Steckzyklus einer Karte und wird beim Stecken einer Karte vom Konnektor generiert. Um eine Karte über mehreren Steckzyklen zu identifizieren kann die Seriennummer der Karte (ICCSN) verwendet werden.

Die über den Konnektor verfügbaren SM-Bs und HBAs, ihre Handles und ICCSNs können über die `GetCards` Operation des Konnektors ermittelt werden.

3.5.1 Erstellung der digitalen Signatur einer Nachricht mit einer SM-B

Das Signieren von ausgehenden Nachrichten erfolgt mit dem Schlüssel PrK.HCI.OSIG der SM-B, die der Institution des Senders entspricht. Ein Konnektor kann von mehreren Institutionen (Mandaten) gleichzeitig benutzt werden und dementsprechend mit mehreren SM-Bs, die den unterschiedlichen Identitäten entsprechen, ausgestattet sein. Die Ermittlung der SM-B, die für die Erstellung der Nachrichtensignatur verwendet werden soll, kann entsprechend dem in Abbildung 12 dargestellten Aktivitätsdiagramm erfolgen. Die Aktivitäten und deren Reihenfolge haben illustrativen und nicht normativen Charakter. Die konkrete Umsetzung kann sich unterscheiden, solange das Ergebnis das Gleiche ist.

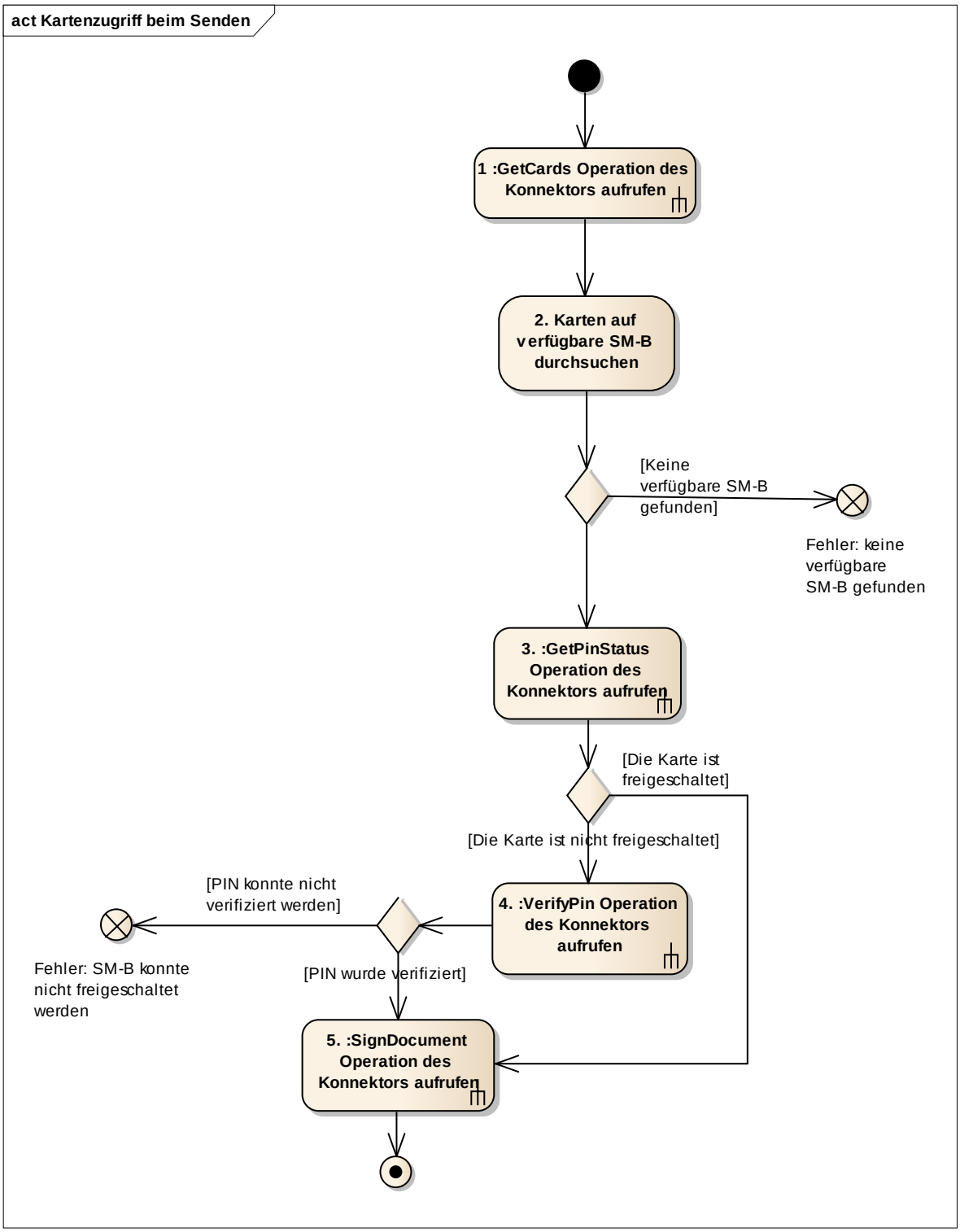

Abbildung 12: Abb_Zugriff_SMB SM-B-Zugriff zur Erstellung der Nachrichtensignatur

Es folgt die Beschreibung der einzelnen Aktivitäten des Diagramms:

1. Die über den Konnektor verfügbaren Karten werden über die Operation `GetCards` mit dem Parameter `Context` (dem Sender entsprechender Aufrufkontext aus dem Benutzernamen) ermittelt.

2. In den anhand des Aufrufkontexts über `GetCards` ermittelten Karten wird nach einer verfügbaren SM-B gesucht:
   - Falls eine verfügbare SM-B gefunden wurde, wird mit Aktivität 3 fortgesetzt.
   - Falls sich unter den verfügbaren Karten keine SM-B befindet, kann die Nachricht nicht signiert werden und das Senden wird abgebrochen.
3. Um festzustellen, ob die Eingabe der PIN für die Freischaltung der Karte notwendig ist, wird die `GetPinStatus` Operation des Konnektors aufgerufen. Dabei werden die Parameter `Context` (dem Sender entsprechender Aufrufkontext), `CardHandle` (Handle der ausgewählten SM-B) und `PinTyp` (PIN.SMC) verwendet.
   - Falls die Karte freigeschaltet ist, fährt das Clientmodul mit Aktivität 5 fort.
   - Falls eine PIN-Eingabe erforderlich ist, fährt das Clientmodul mit Aktivität 4 fort.
4. Für die Eingabe der PIN zur Freischaltung der ausgewählten Karte wird die `VerifyPin` Operation des Konnektors verwendet. Die Operation wird mit den Parametern `Context` (dem Sender entsprechender Aufrufkontext), `CardHandle` (Handle der ausgewählten SM-B), `PinTyp` (PIN.SMC) aufgerufen. Der Sender wird zur Eingabe der PIN über das Display des Kartenterminals angefordert.
5. Die Signatur der KOM-LE-Nachricht erfolgt unter Verwendung der `SignDocument` Operation des Konnektors. Dabei werden die Parameter `Context` (dem Sender entsprechender Aufrufkontext), `CardHandle` (Handle der ausgewählten SM-B), `KeyReference` (C.OSIG) verwendet. Die Verwendung weiterer Parameter muss unter Berücksichtigung der Anforderungen aus [gemSMIME_KOMLE] erfolgen.

⌦ **KOM-LE-A_2052 Quellen zur Ermittlung der SM-B des Senders beim Signieren**

Das Clientmodul MUSS die Menge der verfügbaren Karten, die über die Operation GetCards des Konnektors anhand des Aufrufkontexts des Senders ermittelt werden, nach einer verfügbaren SM-B durchsuchen. ⌫

⌦ **KOM-LE-A_2057 Abbrechen des Signierens, wenn keine SM-B verfügbar ist**

Das Clientmodul MUSS das Signieren einer Nachricht abbrechen, wenn für die Erstellung der Signatur keine SM-B verfügbar/gesteckt ist. ⌫

⌦ **KOM-LE-A_2058 Abbrechen des Signierens, wenn Freischaltung der erforderlichen SM-B fehlschlägt**

Das Clientmodul MUSS das Signieren einer Nachricht abbrechen, wenn die Freischaltung der für die Erstellung der Signatur erforderlichen SM-B fehlschlägt. ⌫

3.5.2 Prüfung der digitalen Signatur einer Nachricht

Die Prüfung der digitalen Signatur einer Nachricht erfolgt mittels der `VerifyDocument` Operation des Konnektors. Dabei werden die Parameter `Context` (dem Empfänger entsprechender Aufrufkontext) und `Document` (signierte Daten) verwendet.

3.5.3 Verschlüsselung einer Nachricht

Die Verschlüsselung einer Nachricht erfolgt mittels der `EncryptDocument` Operation des Konnektors. Dabei werden die Parameter `Context` (dem Empfänger entsprechender Aufrufkontext), `Document` (zu verschlüsselnde Daten) und `Certificate` (alle Zertifikate mit denen die Nachricht verschlüsselt werden soll) verwendet.

3.5.4 Entschlüsselung einer Nachricht mit einer SM-B bzw. einem HBA

Für die Entschlüsselung von empfangenen Nachrichten verwendet das Clientmodul den privaten Schlüssel PrK.HP.ENC eines HBA bzw. den privaten Schlüssel PrK.HCI.ENC einer SM-B. Die Zuordnung von den für die Verschlüsselung verwendeten Zertifikaten und den E-Mail-Adressen der Empfänger wird im `recipient-emails` Attribut des CMS-Objektes mit den verschlüsselten Daten abgebildet (siehe [gemSMIME_KOMLE]). Die Ermittlung des HBAs bzw. der SM-B, die für die Entschlüsselung der empfangenen Nachricht verwendet wird, kann entsprechend dem in Abbildung 13 dargestellten Aktivitätsdiagramm durchgeführt werden. Die Aktivitäten und deren Reihenfolge haben illustrativen und nicht normativen Charakter. Die konkrete Umsetzung kann sich unterscheiden, solange das Ergebnis das Gleiche ist.

© gematik - öffentlich

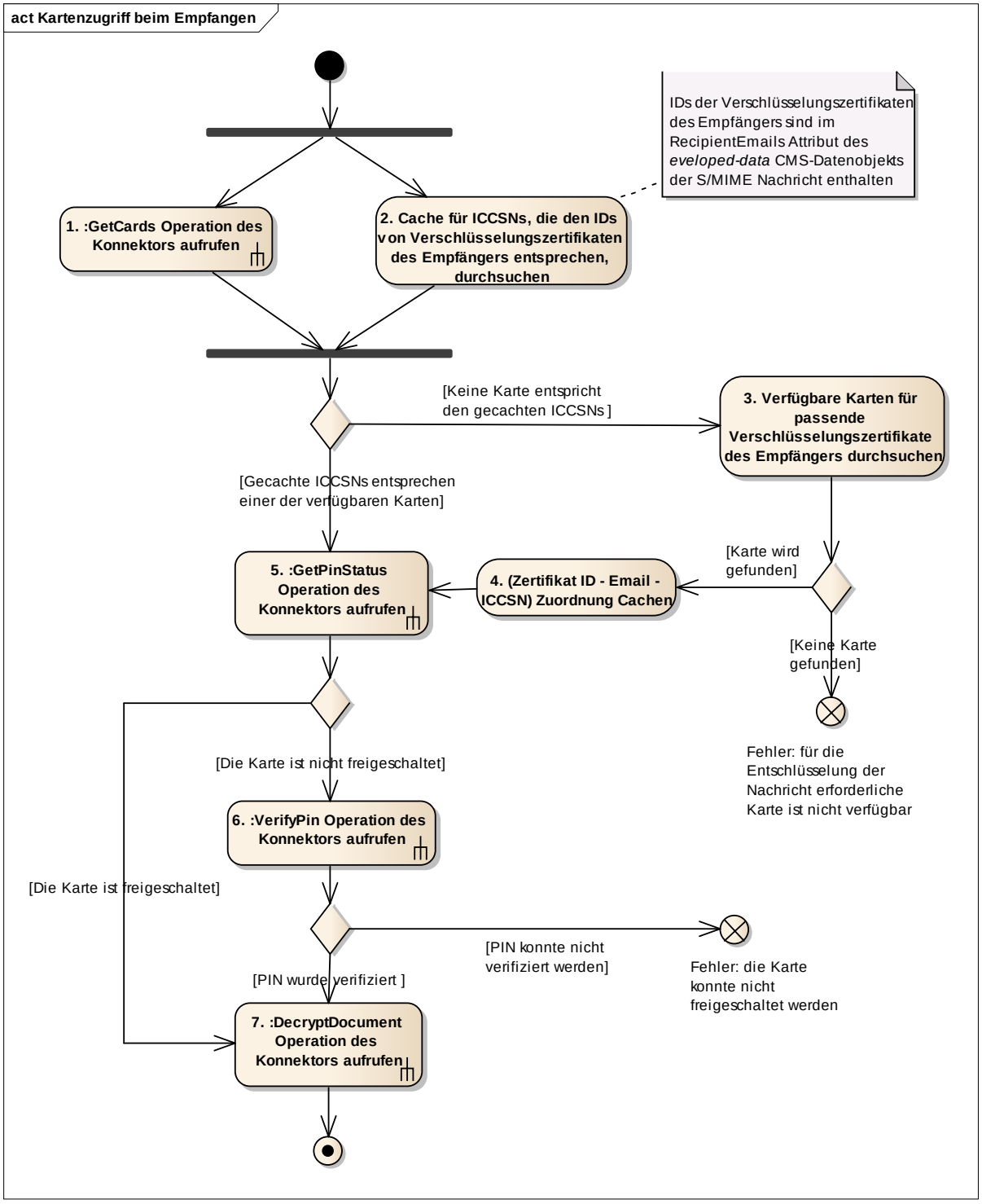

Abbildung 13: Abb_Zugriff_SMB_HBA SM-B/HBA-Zugriff zur Nachrichtentschlüsselung

Es folgt die Beschreibung der einzelnen Aktivitäten des Diagramms:

1. Die über den Konnektor verfügbaren Karten werden über die Operation `GetCards` mit dem Parameter `Context` (dem Empfänger entsprechender Aufrufkontext) ermittelt.

2. Um die Anzahl der Zugriffe auf die Schnittstellen des Konnektors zu reduzieren, verwaltet das Clientmodul einen Cache, der Zuordnungen zwischen E-Mail-Adresse, Zertifikats-ID und ICCSN von HBA/SM-B zwischenspeichert. Dabei sind die gespeicherten Zertifikats-IDs vom ASN.1-Typ `IssuerAndSerialNumber` (siehe [gemSMIME_KOMLE#2.3.3]). Der Cache wird anhand der E-Mail-Adresse des Empfängers und der zugehörigen Zertifikats-IDs aus dem `recipient-emails` Attribut des CMS-Objektes durchsucht.
   - Falls ein passender Eintrag im Cache gefunden wird und die ICCSN dieses Eintrages mit einer über `GetCards` ermittelten ICCSN übereinstimmt, fährt das Clientmodul mit Aktivität 5 fort.
   - Falls der Cache keine passenden Einträge enthält, fährt das Clientmodul mit Aktivität 3 fort.
3. Die IDs der Verschlüsselungszertifikate (Ermittlung über die Operation `ReadCardCertificate` des Konnektors) der über `GetCards` ermittelten HBAs und SM-Bs werden mit den Zertifikats-IDs aus dem `recipient-emails` Attribut des CMS-Objektes, die zur E-Mail-Adresse des Empfängers gehören, verglichen.
   - Falls eine Karte mit passender Zertifikats-ID vorhanden ist, fährt das Clientmodul mit Aktivität 4 fort.
   - Falls keine passende Karte gefunden wird, wird die Entschlüsselung der Nachricht abgebrochen.
4. Die ermittelte (ICCSN – E-Mail-Adresse – Zertifikats-ID) Zuordnung wird im Cache des Clientmoduls gespeichert.
5. Um festzustellen ob die Eingabe der PIN zur Freischaltung der ermittelten Karte notwendig ist, wird die Operation `GetPinStatus` des Konnektors mit den Parametern `Context` (dem Emfänger entsprechender Aufrufkontext), `CardHandle` (Handle der SM-B bzw. des HBA), `PinTyp` (PIN.SMC für SM-B bzw. PIN.CH für HBA) aufgerufen.
   - Falls die Karte freigeschaltet ist, fährt das Clientmodul mit Aktivität 7 fort.
   - Falls die PIN-Eingabe erforderlich ist, fährt das Clientmodul mit Aktivität 6 fort.
6. Die Operation `VerifyPin` des Konnektors wird mit den Parametern `Context` (dem Empfänger entsprechender Aufrufkontext), `CardHandle` (Handle der/des ausgewählten SM-B/HBA), `PinTyp` (PIN.SMC für SM-B bzw. PIN.CH für HBA) aufgerufen. Der Empfänger wird zur Eingabe der PIN über das Display des Kartenterminals aufgefordert.
7. Die Operation `DecryptDocument` des Konnektors wird mit den Parametern `Context` (dem Empfänger entsprechender Aufrufkontext), `CardHandle` (Handle der SM-B bzw. des HBA), `KeyReference` (C.ENC), `Document` (die verschlüsselten Daten) aufgerufen.

⌦ **KOM-LE-A_2059 Verwendung des recipient-emails Attributs beim Entschlüsseln**

Das Clientmodul MUSS die Suche nach der zur Entschlüsselung erforderlichen Karte anhand der E-Mail-Adresse des Empfängers und der zugehörigen Zertifikats-IDs aus dem `recipient-emails` Attribut des CMS-Objektes der KOM-LE-Nachricht durchführen. ⌫

⌦ **KOM-LE-A_2060 Quellen zur Ermittlung der erforderlichen Karte beim Entschlüsseln**

Das Clientmodul MUSS für die Ermittlung der zur Entschlüsselung einer Nachricht erforderlichen Karte primär seinen Cache durchsuchen. Wird die erforderliche Karte nicht über den Cache gefunden, MUSS das Clientmodul die Menge der verfügbaren Karten (wird über die Operation `GetCards` des Konnektors ermittelt) nach der Karte mit dem passenden Verschlüsselungszertifikat (unter Verwendung der Operation `ReadCardCertificate` des Konnektors) durchsuchen. ⌫

⌦ **KOM-LE-A_2061 Speichern von Zuordnungen im Cache beim Entschlüsseln**

Wird beim Entschlüsseln die erforderliche Karte (SM-B bzw. HBA) unter Verwendung der Operation `ReadCardCertificate` des Konnektors ermittelt, MUSS das Clientmodul die zu dieser Karte korrespondierende Zuordnung von E-Mail-Adresse des Empfängers, Zertifikats-ID und ICCSN im Cache speichern. ⌫

⌦ **KOM-LE-A_2062 Abbrechen des Entschlüsseln, wenn die erforderliche Karte nicht verfügbar ist**

Das Clientmodul MUSS die Entschlüsselung einer Nachricht abbrechen, wenn die für die Entschlüsselung erforderliche Karte (SM-B bzw. HBA) nicht verfügbar ist. ⌫

⌦ **KOM-LE-A_2063 Abbrechen des Entschlüsseln, wenn Freischaltung der erforderlichen Karte fehlschlägt**

Das Clientmodul MUSS die Entschlüsselung einer Nachricht abbrechen, wenn die Freischaltung der für die Entschlüsselung erforderlichen Karte fehlschlägt. ⌫

4 Nichtfunktionale Anforderungen

In diesem Kapitel werden nichtfunktionale Anforderungen an das KOM-LE-Clientmodul definiert.

4.1 Transportsicherung

Beim Senden bzw. Empfangen von Nachrichten baut das Clientmodul mit folgenden Systemen Verbindungen auf:

- Clientsysteme (muss stets über TLS erfolgen),
- KOM-LE-Fachdienste (muss stets über TLS erfolgen) und
- Konnektor (kann mit oder ohne TLS erfolgen).

In diesem Kapitel werden die Anforderungen an den Aufbau der TLS-Verbindungen mit diesen Systemen definiert.

4.1.1 Allgemeine Festlegungen

Die Vorgaben zu X.509-Identitäten für die TLS/SSL-Authentifizierung, unterstützten TLS-Versionen und TLS Cipher Suites werden aus [gemSpec_Krypt] übernommen.

⌦ **KOM-LE-A_2064 Verwendung von X.509-Identitäten bei der TLS-Authentifizierung**

Das Clientmodul KOM-LE MUSS bei der Verwendung von X.509-Identitäten für die TLS-Authentifizierung sowie dem Aufbau von TLS-Verbindungen die Vorgaben aus [gemSpec_Krypt] beachten. ⌫

Der Aufbau von TLS-Verbindungen mit Clientsystemen oder die zertifikatsbasierte clientseitige Authentisierung beim Aufbau von TLS-Verbindungen mit dem Konnektor oder den Fachdiensten erfordert das Vorhandensein des entsprechenden Schlüsselmaterials.

Üblicherweise liegt ein Zertifikat zusammen mit dem zugehörigen geheimen Schlüssel in einem standardisierten und passwortgeschützten Format (p12) [PKCS#12] vor. Das Clientmodul kann ein Zertifikat und den zugehörigen geheimen Schlüssel auf mindestens zwei Arten nutzen:

1. Das Clientmodul importiert das Zertifikat und den Schlüssel aus der p12-Datei und verwaltet diese anschließend in einem eigenen Schlüsselspeicher. Dazu muss während des Importvorgangs das Passwort der p12-Datei eingegeben werden (Transportsicherung). Danach hat das Clientmodul Zugriff auf den für den TLS-Verbindungsaufbau benötigten privaten Schlüssel.
2. Das Clientmodul nutzt einen Systemschlüsselspeicher, z.B. den Zertifikatsspeicher von Windows oder den des Java JRE. Auch hier ist für den

Importvorgang das Passwort des der p12-Datei einzugeben. Anschließend stehen das Zertifikat und der Schlüssel über entsprechende Systemfunktionen/Bibliotheken zur Verfügung. Idealerweise kann der Administrator des Clientmoduls im gewählten Zertifikatsspeicher browsen und das gewünschte Zertifikat für die Verwendung auswählen. Alternativ kann in der Clientmodul-Konfiguration eine eindeutige Referenz auf das Zertifikat (Name oder Index) eingegeben werden.

⌦ **KOM-LE-A_2065 Schutz des Schlüsselspeichers für TLS-Verbindungen**

Das Clientmodul MUSS das für den Aufbau von TLS-Verbindungen mit dem Konnektor und Clientsystemen benötigte Schlüsselmaterial in einem durch Passwort geschützten sicheren Schlüsselspeicher ablegen. ⌫

4.1.2 Transportsicherung zwischen Clientsystem und Clientmodul

Die SMTP, POP3 und LDAP-Verbindungen zwischen dem Clientmodul und den Clientsystemen können über TLS geschützt werden. Um mit marktüblichen E-Mail-Clients kompatibel zu sein, ist die Unterstützung von TLS 1.0 durch das Clientmodul erforderlich. Andererseits wird in [gemSpec_Krypt] die Verwendung von TLS 1.0 verboten. Aus Sicht von KOM-LE ist diese Abweichung zulässig, da die Datenübertragung zwischen den Clientsystemen und dem Clientmodul in der geschützten Leistungserbringerumgebung stattfindet.

⌦ **KOM-LE-A_2066 Verwendung von TLS für SMTP-Verbindungen mit Clientsystemen**

Für SMTP-Verbindungen zwischen Clientsystem und Clientmodul MUSS TLS verwendet werden. ⌫

⌦ **KOM-LE-A_2067 Verwendung von TLS für POP3-Verbindungen mit Clientsystemen**

Für POP3-Verbindungen zwischen Clientsystem und Clientmodul MUSS TLS verwendet werden. ⌫

⌦ **KOM-LE-A_2069 Unterstützung von TLS Version 1.0 für die Kommunikation mit Clientsystemen**

Das Clientmodul MUSS, abweichend von der Anforderung GS-A_4387 aus [gemSpec_Krypt], den Serverteil von TLS Version 1.0 für die Kommunikation mit Clientsystemen unterstützen. Dabei muss sich das Clientmodul gegenüber dem Clientsystem unter Verwendung des vom KOM-LE-Anbieter zur Verfügung gestellten TLS-Server-Zertifikats C.CM.TLS-CS authentifizieren.

Unabhängig von dieser Anforderung MUSS das Clientmodul auch die Anforderungen GS-A_4385 und GS-A_4386 aus [gemSpec_Krypt] erfüllen. ⌫

⌦ **KOM-LE-A_2181 Authentifizierung von Clientsystemen gegenüber dem Clientmodul**

Das Clientmodul MUSS für den Aufbau von TLS-Verbindungen mit den Clientsystemen sowohl die Möglichkeit, die zertifikatsbasierte

Clientauthentifizierung zu verwenden, als auch ohne Clientauthentifizierung zu arbeiten, unterstützen. ⌫

4.1.3 Transportsicherung zwischen Clientmodul und Konnektor

Die Kommunikation zwischen Clientmodul und Konnektor basiert auf HTTP. Der Konnektor bietet vier Varianten der HTTP(S)-Verbindung an:

(1) TLS deaktiviert. Verwendung von HTTP ohne Absicherung auf Transportebene.

(2) TLS ohne Client-Authentifizierung.

(3) TLS mit Client-Authentifizierung. Die Client-Authentisierung muss mit den Zertifikaten erfolgen, die der Administrator entweder mit seinen eigenen Mitteln selbst oder mittels des Konnektors erzeugt. In beiden Fällen müssen diese Zertifikate sowohl im Clientmodul (hier zusammen mit ihren privaten Schlüsseln), als auch im Konnektor vorhanden sein.

(4) Kombination von TLS ohne Client-Authentifizierung und HTTP-Basic-Authentifizierung. Das Clientmodul muss Benutzername und Passwort für die HTTP-Basic-Authentifizierung statisch konfigurieren, so dass eine Übereinstimmung mit der Konfiguration am Konnektor besteht.

Für die Basic-Authentifizierung (auch “Basic Access Authentication”, ein Standard der HTTP-Authentifizierung) soll dabei das Clientmodul die notwendigen Parameter „Benutzername“ und „Passwort“ verwalten. Das Clientmodul muss über entsprechende Konfigurationsparameter verfügen. Diese müssen mit den gleichen Werten für Benutzername und Passwort befüllt werden, wie an der Managementschnittstelle des Konnektors.

Die zertifikatsbasierte Client-Authentifizierung erfolgt mit einem Zertifikat, das im passwortgeschützten Schlüsselspeicher gespeichert wird.

⌦ **KOM-LE-A_2070 Verbindungsaufbau mit dem Konnektor mit TLS**

Das Clientmodul MUSS für Verbindungen mit dem Konnektor immer TLS verwenden. ⌫

⌦ **KOM-LE-A_2071 TLS-Verbindung mit dem Konnektor mit oder ohne zertifikatsbasierter Client-Authentifizierung**

Das Clientmodul MUSS konfigurierbar die Verwendung von TLS mit oder ohne zertifikatsbasierter Client-Authentifizierung für Verbindungen mit dem Konnektor ermöglichen. Standardmäßig muss die zertifikatsbasierte Client-Authentifizierung aktiviert sein. ⌫

⌦ **KOM-LE-A_2072 Verwendung von HTTP-Basic-Authentifizierung für TLS-Verbindungen mit dem Konnektor**

Das Clientmodul MUSS konfigurierbar die Verwendung von HTTP-Basic-Authentifizierung in einem TLS-Kanal für Verbindungen mit dem Konnektor ermöglichen. ⌫

4.1.4 Transportsicherung zwischen Clientmodul und Fachdienst

Die Verbindungen zwischen KOM-LE-Clientmodul und KOM-LE-Fachdiensten sowie zwischen KOM-LE-Clientmodul und Verzeichnisdienst erfolgen immer über TLS. Der TLS Handshake zwischen dem Clientmodul und dem MTA, POP3-Server bzw. Verzeichnisdienst findet unmittelbar nach dem Aufbau der entsprechenden TCP-Verbindung statt. Damit wird sichergestellt, dass die Anmeldungsdaten des Nutzers immer über die mit TLS geschützte Verbindung transportiert werden.

Während des Aufbaus der TLS-Verbindung authentifizieren sich die KOM-LE-Fachdienste bzw. der Verzeichnisdienst gegenüber dem Clientmodul mit X.509 TLS-Server-Zertifikaten. Zur Überprüfung dieser Zertifikate verwendet das Clientmodul die Operation `VerifyCertificate` des Konnektors.

Das Clientmodul wiederum authentisiert sich gegenüber den KOM-LE-Fachdiensten mit dem vom KOM-LE-Anbieter zur Verfügung gestellten TLS-Client-Zertifikat und dem entsprechenden privaten Schlüssel.

⌦ **KOM-LE-A_2074 Verbindung zu KOM-LE-Fachdiensten immer über TLS**

Das Clientmodul MUSS immer TLS mit beidseitiger Authentifizierung über X.509-Zertifikate aus der PKI der TI-Plattform für die Verbindung mit den KOM-LE-Fachdiensten verwenden. Das TLS-Handshake MUSS unmittelbar nach dem Aufbau der TCP-Verbindung initiiert werden. ⌫

⌦ **KOM-LE-A_2075 Prüfung von TLS-Server-Zertifikaten**

Das Clientmodul MUSS für die Prüfung von TLS-Server-Zertifikaten der KOM-LE-Fachdienste und des Verzeichnisdienstes die Operation VerifyCertificate des Konnektors benutzen. ⌫

⌦ **KOM-LE-A_2182 Verwendung des vom KOM-LE-Anbieter zur Verfügung gestellten Zertifikats für die clientseitige TLS-Authentifizierung**

Das Clientmodul MUSS sich mit dem vom KOM-LE-Anbieter zur Verfügung gestellten TLS-Client-Zertifikat C.CM.TLS-CS gegenüber dem Server authentifizieren. ⌫

4.2 Nutzung von Webservice-Schnittstellen des Konnektors

Aus der Herstellerdokumentation des Konnektors ist der FQDN zu entnehmen, unter dem der Konnektor seinen Dienstverzeichnisdienst anbietet. Innerhalb des FQDN können Hostname und Domain-Name je nach Konfiguration der LE-Umgebung individuell konfiguriert sein. Der resultierende FQDN des Dienstverzeichnisdienstes muss in die Konfiguration des Clientmoduls übernommen werden.

Durch das Auslesen des Dienstverzeichnisdienstes erhält das Clientmodul Webservice-Endpunkte von Diensten des Konnektors. Die Dienste des Konnektors sind versioniert. Es ist möglich, dass ein Konnektor mehrere Versionen eines Dienstes gleichzeitig anbietet. Die Versionierung der Dienste hilft dem Clientmodul dabei, genau die Dienstversionen zu nutzen, die es clientseitig implementiert hat.

Da nicht davon ausgegangen werden kann, dass die Inhalte des Dienstverzeichnisdienstes statisch sind, sollte das Lesen des Verzeichnisses beim Programmstart und in Fehlersituationen erfolgen, um den Dienstverzeichnis-Cache zu erneuern. Die weitere Kommunikation mit den Diensten des Konnektors erfolgt dann über die im Dienstverzeichnis-Cache propagierten Dienstendpunkte.

⌦ **KOM-LE-A_2076 Ermittlung der Serviceendpunkte des Konnektors**

Das Clientmodul MUSS die Endpunkte der Services, die der Konnektor anbietet, aus dem Dienstverzeichnisdienst (DVD) ermitteln und die Endpunktinformationen der Dienste lokal cachen. Der DVD ist unter einem FQDN, der im Clientmodul konfiguriert ist, erreichbar. Wenn ein Verbindungsproblem auftritt (Dienst nicht erreichbar), MUSS das Clientmodul einen Refresh auf die Endpunktinformationen des Dienstverzeichnisdienstes durchführen. ⌫

⌦ **KOM-LE-A_2077 Auswahl der unterstützten Version einer Dienstschnittstelle des Konnektors**

Das Clientmodul MUSS in der Lage sein, die von ihm unterstützte Dienstversion unter mehreren vom Konnektor angebotenen Dienstschnittstellen auszuwählen. ⌫

4.3 Protokollierung/Logging

Das Clientmodul soll Protokolldateien schreiben, die eine Analyse technischer Vorgänge erlauben. Diese Protokolldateien sind dafür vorgesehen, aufgetretene Fehler zu identifizieren, die Performance zu analysieren und interne Abläufe zu beobachten. Um die Anforderungen an den Datenschutz zu gewährleisten, dürfen keine medizinischen und personenbezogenen Daten protokolliert werden. Geheimes Schlüsselmaterial darf ebenfalls nicht protokolliert werden.

⌦ **KOM-LE-A_2079 Protokolldateien für Ablauf, Performance und Fehler**

Das Clientmodul MUSS das Protokollieren von Abläufen, Performanceinformationen und Fehlern ermöglichen. ⌫

⌦ **KOM-LE-A_2080 Keine Protokollierung sensibler Daten**

Das Clientmodul DARF medizinische und personenbezogene Daten sowie geheimes Schlüsselmaterial und Passwörter NICHT protokollieren. ⌫

Die Protokolldateien folgen einem einheitlichen Format, das vom Hersteller festgelegt wird. Es muss geeignet sein, automatische Auswertungen mit wenig Aufwand durch Dritte zu ermöglichen. Ein Vorbild ist das Weblog des Apache Webservers.

⌦ **KOM-LE-A_2081 Format der Protokolldateien**

Das KOM-LE-Clientmodul MUSS Protokolldateien in einem einheitlichen Format erstellen, um eine automatisierte Auswertung zu ermöglichen. ⌫

Der Zugriff auf Protokolldateien muss auf autorisierte Personen durch angemessene technische oder organisatorische Maßnahmen eingeschränkt werden. Die Logdateien können auf ein separates Speichermedium kopiert werden. Zudem soll der Administrator das Protokollieren für die Performanceanalyse und der internen Abläufe einzeln

deaktivieren und wieder aktivieren können. Für den Produktivbetrieb soll das Protokollieren der internen Abläufe grundsätzlich deaktiviert sein. Damit die Protokolldateien nur begrenzten Speicherplatz belegen, werden sie automatisch nach einem konfigurierbaren Zeitraum gelöscht bzw. überschrieben.

⌦ **KOM-LE-A_2082 Zugriff auf Protokolldateien einschränken**

Das KOM-LE-Clientmodul MUSS den Zugriff auf Protokolldateien auf autorisierte Personen durch angemessene technische oder organisatorische Maßnahmen einschränken. ⌫

⌦ **KOM-LE-A_2083 Kopien der Protokolldateien**

Das KOM-LE-Clientmodul MUSS autorisiertem Personal das Anfertigen von Kopien der Protokolldateien auf separaten Speichermedien ermöglichen. ⌫

⌦ **KOM-LE-A_2084 Aktivierung und Deaktivierung der Protokollierung von Performanceinformationen**

Das KOM-LE-Clientmodul MUSS das Aktivieren und Deaktivieren der Protokollierung von Performanceinformationen ermöglichen. ⌫

⌦ **KOM-LE-A_2085 Begrenzung des Speicherplatzes für Protokolldateien**

Das KOM-LE-Clientmodul MUSS den verwendeten Speicherplatz für die Protokolldateien begrenzen, indem diese automatisch nach einem konfigurierbaren Zeitraum gelöscht oder überschrieben werden. ⌫

Um mehrere Protokolleinträge zu korrelieren, soll beim Aufruf einer Operation eine Vorgangsnummer gebildet werden. Diese Vorgangsnummer wird in allen Protokolleinträgen dieses Operationsaufrufs genutzt. Die Vorgangsnummer wird vom KOM-LE-Clientmodul pseudozufällig gebildet.

⌦ **KOM-LE-A_2086 Vorgangsnummer für Protokolleinträge**

Das KOM-LE-Clientmodul MUSS eine Vorgangsnummer beim Aufruf einer Operation pseudozufällig bilden, um alle zugehörigen Protokolleinträge zum Operationsaufruf zu korrelieren. ⌫

4.3.1 Ablaufprotokoll

Die Protokolleinträge im Ablaufprotokoll enthalten mindestens die in Tabelle 8 aufgezählten Felder.

Tabelle 8: Tab_Felder_Ablauf_Prot Felder im Ablaufprotokoll

Feld	Beschreibung
Vorgangsnummer	Pseudo-zufällige Zeichenkette zur Korrelation der Protokolleinträge
Zeitpunkt	Zeitpunkt der Erstellung des Protokolleintrags
Beschreibung	Details zum Ausführungsschritt

© gematik - öffentlich

Das Ablaufprotokoll soll die Ausführungsschritte enthalten, die einen Einblick in den internen Ablauf für Administratoren, Anbieter und Tester ermöglichen und die Analyse von Fehlersituationen erleichtern.

⌦ **KOM-LE-A_2087 Felder zur Protokollierung des Ablaufs**

Das KOM-LE-Clientmodul MUSS die Protokollierung des Ablaufs mit mindestens folgenden Feldern ermöglichen:

- pseudozufällige Zeichenkette zur Korrelation der Protokolleinträge,
- Zeitpunkt der Erstellung des Protokolleintrags und
- Details zum Ausführungsschritt. ⌫

4.3.2 Performance

Die Protokolleinträge im Performanceprotokoll enthalten mindestens die in Tabelle 9 aufgezählten Felder und müssen geeignet sein, um die tatsächlichen Ausführungszeiten des KOM-LE-Clientmoduls mit den Vorgaben in Kapitel 4.6.1 zu vergleichen. Für jeden Aufruf einer Schnittstelle des Clientmoduls KOM-LE werden ein oder mehrere Protokolleinträge geschrieben.

Tabelle 9: Tab_Felder_Perf_Prot Felder im Performance-Protokoll

Feld	Beschreibung
Vorgangsnummer	Pseudozufällige Zeichenkette zur Korrelation der Protokolleinträge
Name der Aktion	Name der Aktion für Protokolleintrag
Startzeitpunk	Startzeitpunkt der Aktion
Endezeitpunkt	Endetzeitpunkt der Aktion
Dauer in ms	Dauer in ms

⌦ **KOM-LE-A_2088 Felder zur Protokollierung der Performance**

Das KOM-LE-Clientmodul MUSS die Protokollierung der Performance mit mindestens folgenden Feldern ermöglichen:

- pseudozufällige Zeichenkette zur Korrelation der Protokolleinträge,
- Name der Aktion für den Protokolleintrag,
- Startzeitpunkt der Aktion,
- Endetzeitpunkt der Aktion und
- Dauer in ms. ⌫

Jede der in Tabelle 10 aufgelisteten Aktionen führt zu einem Eintrag im Performanceprotokoll. Diese Durchlaufzeiten sollen separat protokolliert werden, damit die Ausführungszeit des Clientmoduls ohne Zeiten anderer Komponenten ermittelbar ist.

Tabelle 10: Tab_Auslöser_Prot_Entry Auslöser Protokolleinträge im Performanceprotokoll

Auslöser	Name der Aktion für Protokolleintrag	Beschreibung
Ankommen einer SMTP bzw. POP3-Meldung	SMTP bzw. POP3-Meldung	Wird beim Ankommen einer SMTP bzw. POP3-Meldung ausgelöst und endet mit der Weiterleitung an den Fachdienst oder der Antwort an das Clientsystem.
Aufruf einer Operation des Konnektors	Name der Operation	Wird durch den Aufruf einer Operation des Konnektors ausgelöst und endet mit der Rückkehr der Aktion

⌦ **KOM-LE-A_2089 Aktionen zur Protokollierung der Performance**

Das KOM-LE-Clientmodul MUSS für die folgenden Aktionen Einträge in das Performanceprotokoll schreiben:

- Ankommen einer SMTP bzw. POP3-Meldung und
- Aufruf einer Schnittstelle des Konnektors. ⌫

4.3.3 Fehler

Tritt innerhalb einer Operation ein Fehler auf bzw. wird eine Operation nicht beendet, soll trotzdem ein Protokolleintrag erstellt werden, in dem eindeutig auswertbar ist, dass die Ausführung der Operation fehlerhaft war.

Die Protokolleinträge im Fehlerprotokoll enthalten mindestens die in Tabelle 11 aufgezählten Felder.

Tabelle 11: Tab_Felder_Fehler_Prot Felder im Fehlerprotokoll

Feld	Beschreibung
Vorgangsnummer	Pseudozufällige Zeichenkette zur Korrelation der Protokolleinträge
Zeitpunkt	Zeitpunkt der Erstellung des Protokolleintrags
Fehlerdetails	Weiterführende Details zur Fehlermeldung

⌦ **KOM-LE-A_2090 Felder zur Protokollierung der Fehler**

Das KOM-LE-Clientmodul MUSS die Protokollierung von Fehlern mit mindestens folgenden Feldern ermöglichen:

- pseudozufällige Zeichenkette zur Korrelation der Protokolleinträge,

- Zeitpunkt der Erstellung des Protokolleintrags und
- Details zur Fehlermeldung. ⌫

4.4 Konfiguration

Die in der Tabelle 12 aufgeführten Parameter müssen über eine Managementoberfläche oder eine Konfigurationsdatei für das KOM-LE-Clientmodul konfigurierbar sein.

Tabelle 12: Tab_Konf_Param Standardkonfiguration allgemeine Parameter

Parameter	Beschreibung des Parameters	Defaultwert
PORT_SMTP	SMTP-Port für Clientsysteme	25
PORT_POP3	POP3-Port für Clientsysteme	995
TLS_AUTH_KONNEKTOR	Authentifizierung des Clientmoduls gegenüber dem Konnektor bei aktivierter TLS-Verbindung (zertifikatsbasiert, Basic-Athentifizierung, ohne)	zertifikats-basiert
KONNEKTOR_TIMEOUT	Timeout für Aufrufe von Schnittstellen des Konnektors	1 Minute
SMTP_TIMEOUT_SERVER	Timeout für Antworten vom SMTP-Server auf SMTP-Kommandos	5 Minuten
SMTP_TIMEOUT_CLIENT	Timeout für das Warten auf neue SMTP-Kommandos vom Clientsystem	5 Minuten
POP3_TIMEOUT_SERVER	Timeout für Antworten vom POP3-Server auf POP3-Kommandos	5 Minuten
POP3_TIMEOUT_CLIENT	Timeout für das Warten auf neue POP3-Kommandos vom Clientsystem	5 Minuten
TTL_ENC_CERT	Time to Live für gecachte Verschlüsselungs-zertifikate	24 Stunden
TTL_EMAIL_ICCSN	Time to Live für gecachte Zuordnungen von E-Mail-Adressen der Sender bzw. Empfänger zu ICCSNs von deren HBAs/SM-Bs	30 Tage
TTL_PROTS	Time to Live für Protokolldateien.	30 Tage
PROT_PERF	Protokolldatei für Performance	JA
KONNEKTOR_URI	URI des DVD des Konnektors	-

⌦ **KOM-LE-A_2091 Konfigurationsparameter**

Das KOM-LE-Clientmodul MUSS die in Tabelle Tab_Konf_Param aufgelisteten Parameter ausschließlich dem berechtigten Akteur über eine Managementoberfläche oder eine Konfigurationsdatei zur Konfiguration anbieten. ⌫

⌦ **KOM-LE-A_2184 Standardwerte der Konfigurationsparameter**

Die Konfiguration des Clientmoduls MUSS mit den in Tabelle Tab_Konf_Param Standardkonfiguration allgemeine Parameter definierten Defaultwerten ausgeliefert werden. ⌫

⌦ **KOM-LE-A_2092 Warnhinweis bei Abschaltung der TLS-Client-Parameter**

Verändert der Administrator die Parameter TLS_SMTP_CLIENT oder TLS_POP3_CLIENT auf „NEIN“ MUSS das Clientmodul einen Warnhinweis erzeugen, der den Administrator über die Risiken der TLS-Abschaltung informiert und darlegt, dass in diesem Fall der Leistungserbringer die Verantwortung für die Sicherstellung der vertraulichen Übertragung übernimmt. Der Warnhinweis muss durch den Administrator bestätigt werden. ⌫

4.5 Update-Mechanismen

⌦ **KOM-LE-A_2225 Update-Mechanismen**

Der Hersteller des Clientmoduls MUSS Mechanismen für das Updaten des Clientmoduls zur Verfügung stellen. Diese Mechanismen MÜSSEN es auch ermöglichen, dass die TLS-Zertifikate und das zugehörige Schlüsselmaterial des Clientmoduls auf sichere Art und Weise erneuert werden können. ⌫

4.6 Produktleistungen

4.6.1 Performance

Die durch das Clientmodul einzuhaltenden Performanceanforderungen werden in diesem Dokument nicht betrachtet sondern in [gemSpec_Perf] aufgeführt.

4.6.2 Skalierbarkeit

Das Clientmodul kann in Einzelpraxen, Praxisgemeinschaften, Gemeinschaftspraxen oder in medizinischen Versorgungszentren (MVZ) eingesetzt werden. Zusätzlich ist der Einsatz in Krankenhäusern und Umgebungen der Kostenträger vorgesehen. In diesen Umgebungen sind gleichzeitige Sende- und Abholvorgänge möglich. Das Clientmodul muss in der Lage sein, solche Vorgänge parallel bearbeiten zu können.

Im Rahmen dieser Spezifikation wird gefordert, dass ein KOM-LE-Clientmodul grundsätzlich beliebig viele parallele Sende- und Abholvorgänge unterstützt. Die Anzahl der tatsächlich unterstützten parallelen Aufrufe wird durch die eingesetzte Hardware und Beschränkungen des Herstellers begrenzt.

⌦ **KOM-LE-A_2094 Skalierbarkeit**

Das Clientmodul MUSS gleichzeitig für mehrere Clientsysteme nutzbar sein, wobei die Anzahl der tatsächlich unterstützten parallelen Aufrufe dem Hersteller überlassen ist. ⌫

Anhang A

A1 – Abkürzungen

Kürzel	Erläuterung
AUTH	Authentisierung
CMS	Cryptographic Message Syntax
DER	Distinguished Encoding Rules
DVD	Dienstverzeichnisdienst
FQDN	Fully Qualified Domain Name
HBA	Heilberufsausweis
ICCSN	Integrated Circuit Card Serial Number
ID	Identifier
KOM-LE	Kommunikation für Leistungserbringer
LDAP	Leightweight Directory Access Protocol
LE	Leistungserbringer
MTA	Mail Transfer Agent
MUA	Mail User Agent
OCSP	Online Certificate Status Protocol
PIN	Personal Identification Number
POP3	Post Office Protocol Version 3
S/MIME	Secure/Multipurpose Internet Mail Extensions
SMTP	Simple Mail Transfer Protocol
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TI	Telematikinfrastruktur
TLS	Transport Layer Security
URL	Uniform Resource Locator
VZD	Verzeichnisdienst

A2 – Glossar

Das Glossar wird als eigenständiges Dokument, vgl [gemGlossar] zur Verfügung gestellt.

A3 – Abbildungsverzeichnis

Abbildung 1: Abb_Dok_Hierarchie Dokumentenhierarchie KOM-LE 7
Abbildung 2: Abb_KOMLE_Komp KOM-LE-Komponenten 9
Abbildung 3: Abb_Struk_KOMLE_Msg Struktur einer KOM-LE-Nachricht.....10
Abbildung 4: Abb_Send_Msg Senden von Nachrichten 12
Abbildung 5: Abb_State_CM_Send Zustände Clientmodul beim Senden von Nachrichten 13
Abbildung 6: Abb_MTA_Nutzername Format des SMTP- Benutzernamens 16
Abbildung 7: Abb_Sig_Verschl Signieren und Verschlüsseln entsprechend S/MIME Profil 21
Abbildung 8: Abb_Verschl_Msg Verschlüsselung einer Nachricht 27
Abbildung 9: Abb_Empfangen_Msg Empfangen von Nachrichten 32
Abbildung 10: Abb_Status_CM_Empfang Zustände Clientmodul beim Nachrichtenempfang.....33
Abbildung 11: Abb_POP3_Nutzer_Name Format des POP3- Benutzernamens 36
Abbildung 12: Abb_Zugriff_SMB SM-B-Zugriff zur Erstellung der Nachrichtensignatur...51
Abbildung 13: Abb_Zugriff_SMB_HBA SM-B/HBA-Zugriff zur Nachrichtentschlüsselung54

A4 – Tabellenverzeichnis

Tabelle 1: Tab_SMTP_Ant_Init Antworten Clientmodul im CONNECT-Zustand.....14
Tabelle 2: Tab_SMTP_Verbindung SMTP-Antwortcodes für MTA-Verbindungsaufbau ..17
Tabelle 3: Tab_POP3_Ant_Init Antworten Clientmodul im CONNECT-Zustand 34
Tabelle 4: Tab_POP3_Verbindung Antwortcodes für POP3-Server-Verbindungsaufbau 37
Tabelle 5: Tab_Fehlertext_Entschl Fehlertexte für Entschlüsselungsfehler.....43
Tabelle 6: Tab_Strukt_Sig_Prüf_Report Struktur Signaturprüfbericht 44
Tabelle 7: Tab_Verm_Sig_Prüf Vermerke mit Ergebnissen der Signaturprüfung 45
Tabelle 8: Tab_Felder_Ablauf_Prot Felder im Ablaufprotokoll 62
Tabelle 9: Tab_Felder_Perf_Prot Felder im Performance-Protokoll 63
Tabelle 10: Tab_Auslöser_Prot_Entry Auslöser Protokolleinträge im Performanceprotokoll.....64
Tabelle 11: Tab_Felder_Fehler_Prot Felder im Fehlerprotokoll 64
Tabelle 12: Tab_Konf_Param Standardkonfiguration allgemeine Parameter 65

A5 - Referenzierte Dokumente

A5.1 – Dokumente der gematik

Die nachfolgende Tabelle enthält die Bezeichnung der in dem vorliegenden Dokument referenzierten Dokumente der gematik zur Telematikinfrastruktur. Der mit der vorliegenden Version korrelierende Entwicklungsstand dieser Konzepte und Spezifikationen wird pro Release in einer Dokumentenlandkarte definiert, Version und Stand der referenzierten Dokumente sind daher in der nachfolgenden Tabelle nicht aufgeführt. Deren zu diesem Dokument passende jeweils gültige Versionsnummer entnehmen Sie bitte der aktuellsten, auf der Internetseite der gematik veröffentlichten Dokumentenlandkarte, in der die vorliegende Version aufgeführt wird.

[Quelle]	Herausgeber: Titel
[gemGlossar]	gematik: Glossar der Telematikinfrastruktur
[gemLH_KOM-LE]	gematik: Lastenheft Adressierte Kommunikation Leistungserbringer
[gemSpec_FD_KOMLE]	gematik: Spezifikation Fachdienst KOM-LE
[gemSpec_Kon]	gematik: Spezifikation Konnektor
[gemSpec_Krypt]	gematik: Verwendung kryptographischer Algorithmen in der Telematikinfrastruktur
[gemSpec_PKI]	gematik: Spezifikation PKI
[gemSMIME_KOMLE]	gematik: KOM-LE S/MIME Profil 1.0
[gemSysL_KOMLE]	gematik: Systemspezifisches Konzept KOM-LE

A5.2 – Weitere Dokumente

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[RFC1939]	RFC 1939: Post Office Protocol – Version 3, J. Myers, M. Rose, Mai 1996
[RFC2045]	RFC 2045: Multipurpose Internet Mail Extension (MIME) Part One: Format of Internet Message Bodies, N. Freed, N. Borenstein, November 1996
[RFC2046]	RFC 2046: Multipurpose Internet Mail Extension (MIME) Part Two: Media Types, N. Feed, N. Borenstein, November 1996
[RFC2449]	RFC 2449: POP3 Extension Mechanism, R. Gellens, C. Newman, L. Lundblade, November 1998
[RFC3463]	RFC 3463: Enhanced Mail System Status Codes, G. Vaudreuil, Januar 2003
[RFC3464]	RFC 3464: An Extensible Message Format for Delivery Status Notifications, K. Moore, G. Vaudreuil, Januar 2003
[RFC4616]	RFC 4616: The PLAIN Simple Authentication and Security Layer (SASL) Mechanism, K. Zeilenga, August 2006
[RFC4954]	RFC 5248: SMTP Service Extension for Authentication, R. Siemborski, A. Melnikov, März 2007
[RFC5321]	RFC 5321: Simple Mail Transfer Protocol, J. Klensin, Oktober 2008

[Quelle]	Herausgeber (Erscheinungsdatum): Titel
[RFC5322]	RFC 5322: Internet Message Format, P. Resnick, Ed., Oktober 2008
[RFC5750]	RFC 5750: Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Certificate Handiling, B. Ramsdell, S. Turner, Januar 2010
[RFC5751]	RFC 5751: Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification, B. Ramsdell, S. Turner, Januar 2010