# Änderungen in gemSpec_Autorisierung, gemSpec_Dokumentenerwaltung und gemSpec_ePA_FdV wie im folgenden Sequenzdiagramm dargestellt

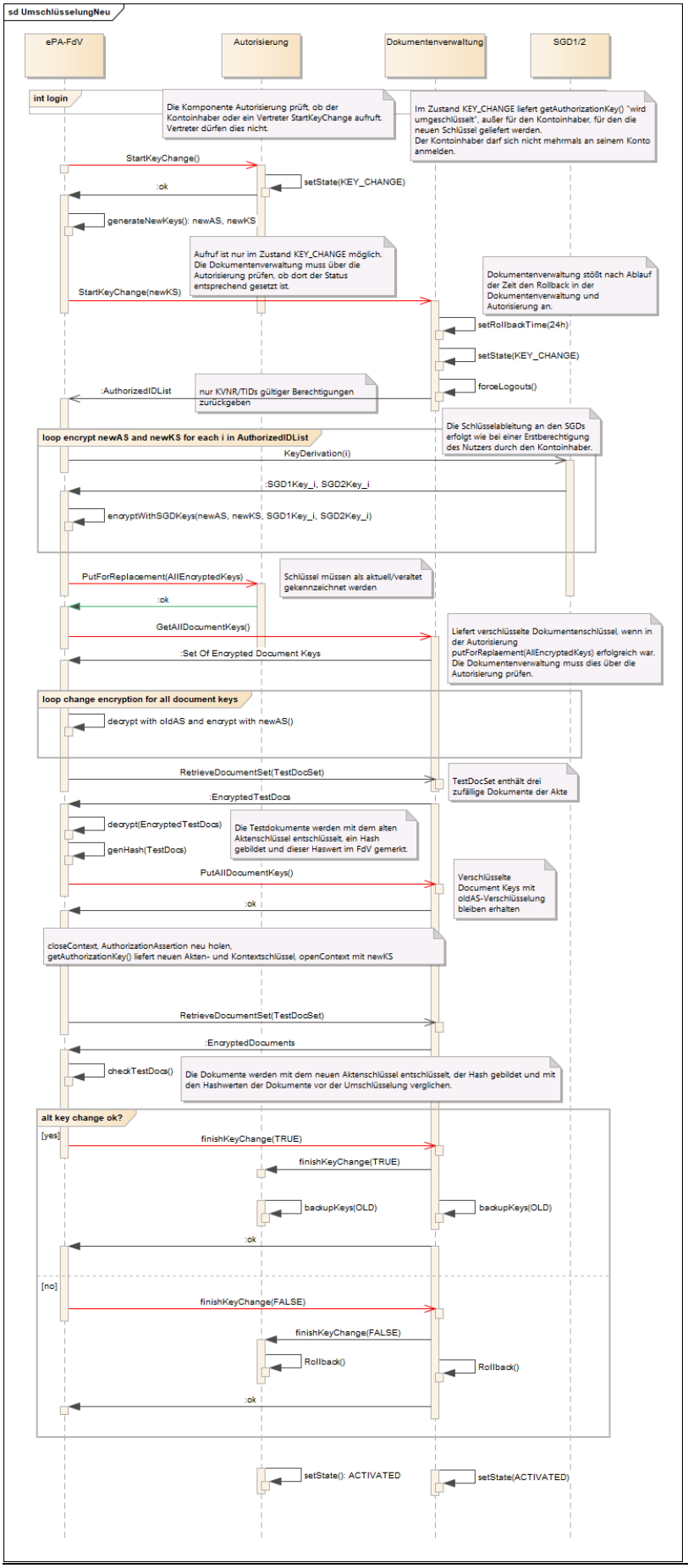
sd UmschlüsselungNeu
ePA-FdV
Autorisierung
Dokumentenverwaltung
SGD1/2
int login
Die Komponente Autorisierung prüft, ob der Kontoinhaber oder ein Vertreter StartKeyChange aufruft. Vertreter dürfen dies nicht.
Im Zustand KEY_CHANGE liefert getAuthorizationKey() "wird umgeschlüsselt", außer für den Kontoinhaber, für den die neuen Schlüssel geliefert werden.
Der Kontoinhaber darf sich nicht mehrmals an seinem Konto anmelden.
StartKeyChange()
setState(KEY_CHANGE)
:ok
generateNewKeys(): newAS, newKS
Aufruf ist nur im Zustand KEY_CHANGE möglich. Die Dokumentenverwaltung muss über die Autorisierung prüfen, ob dort der Status entsprechend gesetzt ist.
Dokumentenverwaltung stößt nach Ablauf der Zeit den Rollback in der Dokumentenverwaltung und Autorisierung an.
StartKeyChange(newKS)
setRollbackTime(24h)
setState(KEY_CHANGE)
forceLogouts()
:AuthorizedIDList
nur KVNR/TIDs gültiger Berechtigungen zurückgeben
Die Schlüsselableitung an den SGDs erfolgt wie bei einer Erstberechtigung des Nutzers durch den Kontoinhaber.
loop encrypt newAS and newKS for each i in AuthorizedIDList
KeyDerivation(i)
:SGD1Key_i, SGD2Key_i
encryptWithSGDKeys(newAS, newKS, SGD1Key_i, SGD2Key_i)
PutForReplacement(AllEncryptedKeys)
Schlüssel müssen als aktuell/veraltet gekennzeichnet werden
:ok
GetAllDocumentKeys()
Liefert verschlüsselte Dokumentenschlüssel, wenn in der Autorisierung putForReplacement(AllEncryptedKeys) erfolgreich war. Die Dokumentenverwaltung muss dies über die Autorisierung prüfen.
:Set Of Encrypted Document Keys
loop change encryption for all document keys
decrypt with oldAS and encrypt with newAS()
RetrieveDocumentSet(TestDocSet)
TestDocSet enthält drei zufällige Dokumente der Akte
:EncryptedTestDocs
decrypt(EncryptedTestDocs)
genHash(TestDocs)
Die Testdokumente werden mit dem alten Aktenschlüssel entschlüsselt, ein Hash gebildet und dieser Haswert im FdV gemerkt.
PutAllDocumentKeys()
Verschlüsselte Document Keys mit oldAS-Verschlüsselung bleiben erhalten
:ok
closeContext, AuthorizationAssertion neu holen, getAuthorizationKey() liefert neuen Akten- und Kontextschlüssel, openContext mit newKS
RetrieveDocumentSet(TestDocSet)
:EncryptedDocuments
checkTestDocs()
Die Dokumente werden mit dem neuen Aktenschlüssel entschlüsselt, der Hash gebildet und mit den Hashwerten der Dokumente vor der Umschlüsselung verglichen.
alt key change ok?
[yes]
finishKeyChange(TRUE)
finishKeyChange(TRUE)
backupKeys(OLD)
backupKeys(OLD)
:ok
[no]
finishKeyChange(FALSE)
finishKeyChange(FALSE)
Rollback()
Rollback()
:ok
setState(): ACTIVATED
setState(ACTIVATED)

**Abbildung 1: neuer Ablauf Umschlüsselung**