

# VOLKSWAGEN

AKTIENGESELLSCHAFT

**CONFIDENTIAL**  
**VERTRAULICH**

## UNECE Softwareupdate - Allgemeine Diagnoseanforderungen

Ergänzungsspezifikation für Steuergeräte

**Technische Entwicklung, Querschnittslastenheft: LAH.DUM.905.E**

|                    |            |
|--------------------|------------|
| <b>Erstausgabe</b> | 22.11.2019 |
|--------------------|------------|

|                       |            |
|-----------------------|------------|
| <b>Änderungsstand</b> | 18.11.2021 |
|-----------------------|------------|

|                          |     |
|--------------------------|-----|
| <b>Lastenheftversion</b> | 1.4 |
|--------------------------|-----|

|                 |         |
|-----------------|---------|
| <b>Baseline</b> | 6.3 ( ) |
|-----------------|---------|

## Inhaltsverzeichnis

|       |   |    |
|-------|---|----|
| 1     | Allgemeines .....   | 4  |
| 1.1   | Zweck.....  | 4  |
| 1.2   | Abkürzungen und Begriffe.....   | 6  |
| 1.3   | Gültigkeitsbereich.....   | 7  |
| 2     | Anforderungen an Steuergeräte.....  | 8  |
| 2.1   | Steuergeräte in Bestandsarchitekturen und in der E <sup>3</sup> 1.2 Architektur .....   | 8  |
| 2.2   | Neue Steuergeräte .....   | 9  |
| 2.2.1 | DK0/DK1-Systeme .....   | 10 |
| 2.2.2 | DK2/DK2F/DK2FV-Systeme .....  | 11 |
| 2.2.3 | DK3/DK3V/DK4/DK4V-Systeme und SWCL .....  | 12 |
| 2.3   | Weitere Steuergeräte .....  | 13 |
| 3     | Anforderungen zur Absicherung der Fahrzeugdiagnose .....  | 14 |
| 3.1   | SFD-E2E-Absicherung.....  | 14 |
| 3.1.1 | Sonderfall für DK4-low/DK4V-low-Systeme mit unterlagerten DK2/DK2F/DK2FV-Systemen .....   | 15 |
| 3.2   | ECU programming data security .....   | 20 |
| 4     | Integrity Validation Data .....   | 21 |
| 4.1   | Allgemeine Anforderungen.....   | 21 |
| 4.1.1 | Programmierungs-Hashwert .....  | 24 |
| 4.1.2 | Konfigurations-Hashwert.....  | 37 |
| 4.2   | Anforderungen an DK4-Systeme auf Basis Q-LAH 80127 bis Version 4.0.....   | 50 |
| 4.2.1 | Programmierungs-Hashwert .....  | 51 |
| 4.2.2 | Konfigurations-Hashwert.....  | 52 |
| 4.3   | Anforderungen an DK2/DK2F/DK2FV-Systeme .....   | 53 |
| 4.3.1 | Programmierungs-Hashwert .....  | 53 |
| 4.3.2 | Konfigurations-Hashwert.....  | 55 |
| 4.4   | Standardsoftware-Modul für Integrity Validation Data.....   | 56 |
| 4.5   | Ablauf.....   | 57 |
| 4.5.1 | Beispielhaftes Auslesen aller relevanten Identifikationsdaten und Integrity Validation Data eines Diagnose-Servers.....                                   | 57 |
| 4.5.2 | Beispielhaftes Schreiben von Daten in ein SFD-E2E abgesichertes DK4/DK4V bzw. DK3/DK3V System.....  | 64 |
| 4.5.3 | Beispielhafte Programmierung eines DK2F/DK2FV-Systems .....   | 67 |
| 5     | Diagnoseobjekte .....   | 70 |
| 5.1   | Datidentifizier.....  | 71 |
| 5.1.1 | 0xF1A3-VW ECU Hardware Version Number.....  | 72 |
| 5.1.2 | 0x0249-Programming_hash.....  | 73 |
| 5.1.3 | 0x0247-Slave_list_programming_hash.....   | 74 |
| 5.1.4 | 0x0245-Configuration_hash .....   | 75 |
| 5.1.5 | 0x0248-Slave_list_configuration_hash .....  | 76 |
| 5.1.6 | 0xF18F-Regulation_x_software_identification_numbers .....   | 77 |
| 5.1.7 | 0x0250-Integrity_validation_data_configuration_list .....   | 79 |
| 5.1.8 | 0x0251-Write_generic_to_sub_system (Schreiben des Datidentifiziers 0x0250 über das DK4-low/DK4V-low-System zum unterlagerten DK2/DK2F/DK2FV-System) ..... | 81 |
| 5.2   | Routinidentifizier .....  | 84 |
| 5.2.1 | Rücksetzen von Konfigurationsparametern .....   | 85 |
| 5.2.2 | Berechnung Integrity Validation Data.....   | 86 |
| 6     | RxSWIN-spezifische Dokumentation .....  | 99 |
| 6.1   | Daten für die RxSWIN-spezifische Dokumentation eines DK3/DK3V/DK4/DK4V-System .....   | 99 |

|     |  |     |
|-----|--|-----|
| 6.2 | Daten für die RxSWIN-spezifische Dokumentation eines SWCL.....             | 100 |
| 6.3 | Daten für die RxSWIN-spezifische Dokumentation eines DK2/DK2F/DK2FV-System | 101 |
| 7   | Mitgeltende Dokumente und Spezifikationen.....                             | 102 |

# 1 Allgemeines

## 1.1 Zweck

[I: F-LAH\_RxSWIN-7]

Dieses Dokument beschreibt technische Anforderungen an Onboard- sowie Offboard-Systeme zur Einhaltung der "UNECE Regulierung zu Software Updates" der "Working Party on Automated/Autonomous and Connected Vehicles (GRVA)" aus der WP.29 ("Regulation on uniform provisions concerning the approval of software update processes").

[I: F-LAH\_RxSWIN-712]

Jede Software zur Umsetzung von Fahrzeugfunktionen muss entsprechend dem Software Update Management System entwickelt werden. Zusätzlich müssen typgenehmigungspflichtige Fahrzeugfunktionen eine regulierungsbezogene Software Identifikationsnummer (RxSWIN) erhalten. Dies betrifft beispielsweise UNECE- oder GB/T-Regulierungen. Diese Anforderungen sind verpflichtend zu erfüllen, um eine Typgenehmigung zu erhalten.

[I: F-LAH\_RxSWIN-433]

Seitens der Diagnose ergeben sich durch die UNECE Regulierung Software Update folgende Anforderungen:

[I: F-LAH\_RxSWIN-732]

- Authentisches Einbringen von Software (Instruction Code and Data)  
Referenz: UNECE Regulierung zu Software Updates-7.2.1.1. - The authenticity and integrity of software updates shall be protected to reasonably prevent their compromise and reasonably prevent invalid updates.  
Umsetzung: Nutzung von Flashdatensicherheit oder ein alternatives, mindestens gleichwertiges Absicherungsverfahren und Schutz der Fahrzeugdiagnose  
[I: F-LAH\_RxSWIN-733]
- Einführen von RxSWIN für typgenehmigungspflichtige Fahrzeugfunktionen  
Referenz: UNECE Regulierung zu Software Updates-2.2. - "Regulation X Software Identification Number (RXSWIN)" means a dedicated identifier, defined by the vehicle manufacturer, representing information about the type approval relevant software of the Electronic Control System contributing to the Regulation N° x type approval relevant characteristics of the vehicle.  
Umsetzung: Einführung eines RxSWIN-Datidentifizier im Gateway  
[I: F-LAH\_RxSWIN-734]
- Einführen eines Integritätsmerkmals von Software (Instruction Code and Data)  
Referenz: UNECE Regulierung zu Software Updates-7.1.2.3. - For every RxSWIN, there shall be documentation describing the software relevant to the RxSWIN of the vehicle type before and after an update. This shall include information of the software versions and their integrity validation data for all relevant software for each RxSWIN.  
Umsetzung: Einführung von Integrity Validation Data für Programm und Daten

[I: F-LAH\_RxSWIN-552]

Nach der Anwendung der UNECE SW Update Regulierung bei einer 1958er-Vertragspartei (Markt bzw. Land) dürfen bei Fahrzeugen ohne den technischen Voraussetzungen (z. B. RxSWIN, Software Versionen) und einer fehlenden SUMS-Zertifizierung keine Softwareupdates durchgeführt werden. Aufgrund der deshalb nicht möglichen Software Update- und Cyber Security-Typprüfung ist eine Typgenehmigung des Fahrzeugs nicht möglich.

[I: F-LAH\_RxSWIN-430]

Die Zuordnung der Fahrzeugprojekte zu Bestands- E<sup>3</sup> 1.2- oder Neu-Architekturen erfolgt durch den Auftraggeber.

[I: F-LAH\_RxSWIN-432]

Für Fahrzeugprojekte mit einer Software Updates Typgenehmigung gelten folgende Anforderungen:

[I: F-LAH\_RxSWIN-825]

Alle programmier- oder konfigurierbare Steuergeräte:

- ECU programming data security (EPDS) für Software
- Schutz der Fahrzeugdiagnose (SFD-Ende-zu-Ende-Absicherung) für Daten
- Ausnahme DK2-System: Absicherung erfolgt über das übergeordnete DK4-low-System
- Integrity Validation Data für Software und Daten

[I: F-LAH\_RxSWIN-826]

Gateway-Steuergerät zusätzlich:

- Umsetzung Diagnosefilter mit SFD-Zugriffsschutz
- RxSWIN mit SFD-Ende-zu-Ende-Absicherung

## 1.2 Abkürzungen und Begriffe

[I: F-LAH\_RxSWIN-21]

**Tabelle 1-1 Abkürzungen und Begriffe**

| Abkürzung oder Begriff | Bezeichnung  | Bedeutung  |
|------------------------|--|--|
| BoM                    | Bill of Material   | Bauzustandsdokumentation   |
| BSB                    | Bauteil-Sachbearbeiter                                     | Rollenbezeichnung d. AUDI AG, gleichzusetzen mit "BTV - Bauteil-Verantwortlicher" d. Volkswagen AG   |
|                        | Codierung/Coding   | Umsetzung des DataIdentifier "0x0600-VW Coding Value" im Steuergerät.<br>Hinweis : Der Inhalt der Codierung wird PR-Nummern-gesteuert über ein Zieldatencontainer in ein Steuergerät eingebracht. Diese Steuergeräte müssen die IVD für Konfigurationsdaten umsetzen.  |
| C                      | Conditional  | Cvt.=C: In Abhängigkeit von bestimmten Bedingungen zu senden/zu implementieren.  |
| Cvt.                   | Convention   | Implementierungsregeln und Konventionen, die für Parameter eines Services gelten.  |
| EPDS                   | ECU programming data security                              | Generischer Begriff für einen Security-Mechanismus der die Authentizität und Integrität eines Steuergeräte-Programmstands über Diagnose feststellt. Dies kann durch die Umsetzung von Flashdatensicherheit (FDS) oder ein alternatives, mindestens gleichwertiges Absicherungsverfahren erfolgen, welches mit der E/E-Security-Abteilung des Auftraggebers abgestimmt ist. |
| FDS                    | Flashdatensicherheit                                       | Kryptografisches Verfahren zur Absicherung der Flashdaten.   |
| GB/T                   | GB steht für Guobiao, chinesisch für „Nationaler Standard“ | Grundlage für den Produkttest, den das Produkt im Zuge der CCC-Zertifizierung durchlaufen muss.  |
| IVD                    | Integrity Validation Data                                  | Hashwerte über Software (Instruction-Code) und Konfigurationsdaten (Data)  |
| M                      | Mandatory  | Cvt.=M: Verbindlich für die Applikations-Software zu implementieren bzw. immer zu senden (engl. für befehlend).  |
| NoImp                  | No implementation  | Eine Implementierung darf für VOLKSWAGEN AG beauftragte Server nicht erfolgen.   |
| RxSWIN                 | Regulierungsbezogene Software Identifikation               | Software Identifikationsnummer pro betroffener Regulierung.  |
| SFD                    | Schutz der Fahrzeugdiagnose                                | Kryptografisches Verfahren zur Absicherung<br>a) des Zugriffs auf Diagnoseobjekte (SFD-Zugriffsschutz) bzw.<br>b) von Diagnoseinhalten (SFD-Ende-zu-Ende-Absicherung).   |
| SUMS                   | Software Update Management System                          | Stellt die Einhaltung von gesetzlichen Anforderungen bzgl. der Bereitstellung von Software Updates durch entsprechende Prozesse beim Hersteller sicher.  |
| SW                     | Software   | Nach UNECE SU ist mit Software der ausführbarer Code und die Daten (Instruction Code and Data) gemeint.  |
| U                      | User-Optional  | Cvt.= NoImp: Eine Implementierung darf für VOLKSWAGEN AG beauftragte Server nicht erfolgen.  |
| UNECE                  | United Nations Economic Commission for Europe              | Wirtschaftskommission für Europa der Vereinten Nationen  |
| UNECE R SU             | UNECE Regulierung zu Software Update                       | Titel: "Regulation on uniform provisions concerning the approval of software update processes".  |
| ZDC                    | Zieldaten-Container  | XML-Datei, die alle Parameter der Varianten eines Servers enthält, die über PR-Nummern ausgewählt werden.  |
| zGW                    | Zentrales Vernetzungs-Gateway                              | Steuergerät mit zentralem Diagnosezugang, z.B. Gateway, ICAS1, HCP5.   |
| VOLKSWAGEN AG-reserved |  | Die Verwendung ist für zukünftige Anwendungen der VOLKSWAGEN AG reserviert.  |

### 1.3 Gültigkeitsbereich

[allg. Anf.: F-LAH\_RxSWIN-9]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Dieses Dokument ist gültig für alle programmier- und konfigurierbaren Steuergeräte in Bestands- sowie in neuen Architekturen, die über die hier beschriebenen Standard-Update-Verfahren (UDS/LUM) aktualisiert werden können.

[allg. Anf.: F-LAH\_RxSWIN-1128]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Software-Updates im Fahrzeug, die über alternative Update-Verfahren, die nicht in diesem Dokument beschrieben sind, durchgeführt werden sollen und deren Software nicht dem Software Update Management System (SUMS) unterliegen sollen, müssen zumindest die folgenden Anforderungen erfüllen bzw. Nachweise erbringen:

- die erforderlichen Security-Maßnahmen müssen angewendet werden, um die Software authentisch und integer in ein Steuergerät einzubringen und auszuführen.
- die Software unterliegt dem Versions- und Konfigurationsmanagement und die Software Updates werden in geeigneter Weise dokumentiert. (freigegebene Softwarestände/Verbünde müssen identifizierbar sein)
- die aktuell im Steuergerät installierte Software muss über Standard Diagnose-Verfahren eindeutig identifizierbar sein.
- für Software die nicht dem SUMS unterliegt muss der Nachweis erbracht werden, dass die Software nicht national/regional typgenehmigungspflichtig ist und keine homologations-relevante Fahrzeugfunktion beeinflussen kann.
- Nachweise sind für die Kommunikation mit den lokalen Genehmigungsbehörden zu dokumentieren.

[allg. Anf.: F-LAH\_RxSWIN-550]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Bei abweichenden Anforderungen zwischen diesem Dokument und den anderen Diagnose-Querschnittslastenheften gelten die Anforderungen aus diesem Lastenheft.

[I: F-LAH\_RxSWIN-521]

Die Farbkennzeichnung der Tabellen dient nur der Lesbarkeit.

[allg. Anf.: F-LAH\_RxSWIN-918]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die als "Prozess-Anforderung" gekennzeichneten Anforderungen sind im Rahmen der Steuergeräteentwicklung nicht zu berücksichtigen.

## 2 Anforderungen an Steuergeräte

### 2.1 Steuergeräte in Bestandsarchitekturen und in der E<sup>3</sup> 1.2 Architektur

[I: F-LAH\_RxSWIN-987]

Die Maßnahmen für Steuergeräte in Bestandsarchitekturen und in der E<sup>3</sup> 1.2 Architektur sind in Abstimmung mit der Sicherheits- (Safety und Security) und der Diagnoseabteilung des Auftraggebers festzulegen und im BT-LAH zu dokumentieren.

[allg. Anf.: F-LAH\_RxSWIN-988]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Umsetzung der Maßnahmen erfolgt nach den Kapiteln

- Neue Steuergeräte
- Anforderungen zur Absicherung der Fahrzeugdiagnose
- Integrity Validation Data
- Diagnoseobjekte

aus diesem Dokument.



## 2.2 Neue Steuergeräte

[I: F-LAH\_RxSWIN-917]

**Tabelle 2-1 Übersicht der Anforderungen aus Sicht Diagnoseklasse**

|   | Zentrales Gateway | DK4(V)          | DK3(V)          | DK2F(V)             | DK2(V)              | DK1(V)/0 |
|---|-------------------|-----------------|-----------------|---------------------|---------------------|----------|
| <b>RxSWIN DataIdentifier</b>  | X                 | NoImp           | NoImp           | NoImp               | NoImp               | NoImp    |
| <b>Diagnostic filter (OBD-port)</b>   | X                 | NoImp           | NoImp           | NoImp               | NoImp               | NoImp    |
| <b>ECU programming data security (EPDS)</b>   | X                 | X               | X <sup>3)</sup> | X                   | NoImp               | NoImp    |
| <b>Protection of vehicle diagnosis (PVD/SFD)</b><br>- SFD Authentication<br>- SFD End2End | X                 | X               | X <sup>1)</sup> | NoImp <sup>2)</sup> | NoImp <sup>2)</sup> | NoImp    |
| <b>Integrity validation data (IVD)</b>  | X                 | X <sup>3)</sup> | X <sup>3)</sup> | X <sup>3)</sup>     | X <sup>3)</sup>     | NoImp    |

[I: F-LAH\_RxSWIN-922]

x = beinhaltet

[I: F-LAH\_RxSWIN-919]

1) = Wenn keine Konfiguration mittels ZDC in der Applikation möglich ist, dann ist der Einsatz von SFD abhängig von der Risiko- /Securityanalyse.

[I: F-LAH\_RxSWIN-923]

2) = erhalten die Daten ohne Signatur vom DK4

[I: F-LAH\_RxSWIN-943]

3) = nur bei bedatbaren (programmier- und/oder konfigurierbaren) Diagnose-Servern

### **2.2.1 DK0/DK1-Systeme**

[I: F-LAH\_RxSWIN-709]

DK0/DK1-Systeme sind von den Anforderungen in diesem Dokument nicht betroffen, da diese nicht bedatbar sind.

## 2.2.2 DK2/DK2F/DK2FV-Systeme

[I: F-LAH\_RxSWIN-941]

Nicht bedatbare (weder programmier- noch konfigurierbare) DK2/DK2F/DK2FV-Systeme sind von diesem Dokument nicht betroffen.

### 2.2.2.1 Programmierbare DK2F/DK2FV-Systeme

[allg. Anf.: F-LAH\_RxSWIN-850]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Programmierbare DK2F/DK2FV-Systeme müssen

[allg. Anf.: F-LAH\_RxSWIN-754]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- ECU programming data security (EPDS) unterstützen.

[allg. Anf.: F-LAH\_RxSWIN-847]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Integrity Validation Data für die Programmierung unterstützen. Siehe Kapitel "Integrity Validation Data".

### 2.2.2.2 Konfigurierbare DK2/DK2F/DK2FV-Systeme

[allg. Anf.: F-LAH\_RxSWIN-851]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Konfigurierbare DK2/DK2F/DK2FV-Systeme

[allg. Anf.: F-LAH\_RxSWIN-755]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- werden per ZDC konfiguriert.

[allg. Anf.: F-LAH\_RxSWIN-756]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- erhalten Daten ohne Signatur vom DK4-low-System. Das DK4-low-Syst führt die SFD-E2E-Signaturprüfung für das DK2/DK2F/DK2FV-Syste durch.

[allg. Anf.: F-LAH\_RxSWIN-757]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- müssen Integrity Validation Data für die Konfiguration unterstützen. Siehe Kapitel "Integrity Validation Data".

## 2.2.3 DK3/DK3V/DK4/DK4V-Systeme und SWCL

[I: F-LAH\_RxSWIN-942]

Nicht bedatbare (weder programmier- noch konfigurierbare) DK3/DK3V/DK4/DK4V-Systeme und SWCL sind von diesem Dokument nicht betroffen.

### 2.2.3.1 Programmierbare DK3/DK3V/DK4/DK4V-Systeme und SWCL

[allg. Anf.: F-LAH\_RxSWIN-852]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Programmierbare DK3/DK3V/DK4/DK4V-Systeme und SWCL müssen

[allg. Anf.: F-LAH\_RxSWIN-759]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- ECU programming data security (EPDS) unterstützen.

[allg. Anf.: F-LAH\_RxSWIN-849]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Integrity Validation Data für die Programmierung unterstützen. Siehe Kapitel "Integrity Validation Data".

### 2.2.3.2 Konfigurierbare DK3/DK3V/DK4/DK4V-Systeme und SWCL

[allg. Anf.: F-LAH\_RxSWIN-853]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Konfigurierbare DK3/DK3V/DK4/DK4V-Systeme und SWCL

[allg. Anf.: F-LAH\_RxSWIN-760]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- werden per ZDC konfiguriert.

[allg. Anf.: F-LAH\_RxSWIN-761]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- müssen SFD-Ende-zu-Ende-Absicherung (SFD-E2E) unterstützen, wenn sie in der Applikation konfigurierbar sind. Siehe Kapitel "SFD-E2E-Absicherung".

[allg. Anf.: F-LAH\_RxSWIN-762]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- müssen Integrity Validation Data für die Konfiguration unterstützen. Siehe Kapitel "Integrity Validation Data".

## 2.3 Weitere Steuergeräte

[I: F-LAH\_RxSWIN-953]

Steuergeräte dieser Kategorie müssen keine zusätzlichen Anforderungen aus diesem Dokument umsetzen.

### 3 Anforderungen zur Absicherung der Fahrzeugdiagnose

#### 3.1 SFD-E2E-Absicherung

[allg. Anf.: F-LAH\_RxSWIN-82]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Diagnose-Server, die SFD-Ende-zu-Ende-Absicherung gemäß Kapitel "Anforderungen an Steuergeräte" umsetzen, müssen das Dokument /4/ (Q-LAH SFD mindestens Version 2.1 inkl. Errata zu Version 2.1) verwenden.

[allg. Anf.: F-LAH\_RxSWIN-341]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Folgende Datenkategorien (Datenarten) gemäß Dokument /8/, die über den Service WriteData-ByIdentifier (2Ehex) geschrieben werden, sind SFD-Ende-zu-Ende abzusichern und in den Diagnosedaten-Tabellen des BT-LAH entsprechend zu dokumentieren:

[allg. Anf.: F-LAH\_RxSWIN-811]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Codierung (ausstattungsabhängiges Ein/Ausschalten von Teilfunktionen)

[allg. Anf.: F-LAH\_RxSWIN-810]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Fahrzeugparameter (fahrzeugabhängige Parameter und Einstellung) die per Applikations-Datensätze übertragen werden

[allg. Anf.: F-LAH\_RxSWIN-809]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Erstbedatungswerte (einmalig (initial) beschreibbare Defaultwerte, Parameter und Einstellungen)

[Prozess-Anf.: F-LAH\_RxSWIN-1148]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Nicht alle SFD-Ende-zu-Ende abgesicherten Anpassungen müssen im ZDC als Modus-K-Teile enthalten sein. SFD-Ende-zu-Ende abgesicherte Entwickler-/Debug-Schnittstellen bzw. Freischaltungen/Einstellungen dürfen nicht Bestandteil des ZDC sein.

[allg. Anf.: F-LAH\_RxSWIN-993]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Ein Steuergerät mit Datensatzdownload Generation 1 nach Dokument /11/ muss beim Einsatz der SFD-Ende-zu-Ende-Absicherung den Datensatzdownload Generation 2 nach Dokument /7/ umsetzen.

### 3.1.1 Sonderfall für DK4-low/DK4V-low-Systeme mit unterlagerten DK2/DK2F/DK2FV-Systemen

[I: F-LAH\_RxSWIN-932]

Bei den folgenden Abbildungen handelt es sich um beispielhafte Teil-Abläufe, die nicht implementiert werden dürfen.

[I: F-LAH\_RxSWIN-101]

DK2/DK2F/DK2FV-Systeme setzen den Schutz der Fahrzeugdiagnose (SFD) mit der Ende-zu-Ende-Absicherung nicht selbst um.

[allg. Anf.: F-LAH\_RxSWIN-100]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Ein DK4-low/DK4V-low-System muss die, über den Service WriteDataByIdentifier (2Ehex) schreibbaren, Konfigurationsdaten eines unterlagerten DK2/DK2F/DK2FV-Systems entsprechend der Anforderung "F-LAH\_RxSWIN-341" vor nicht autorisiertem Zugriff mittels SFD-Ende-zu-Ende-Absicherung schützen. Abweichungen sind mit der Security-Abteilung des Auftraggebers abzustimmen.

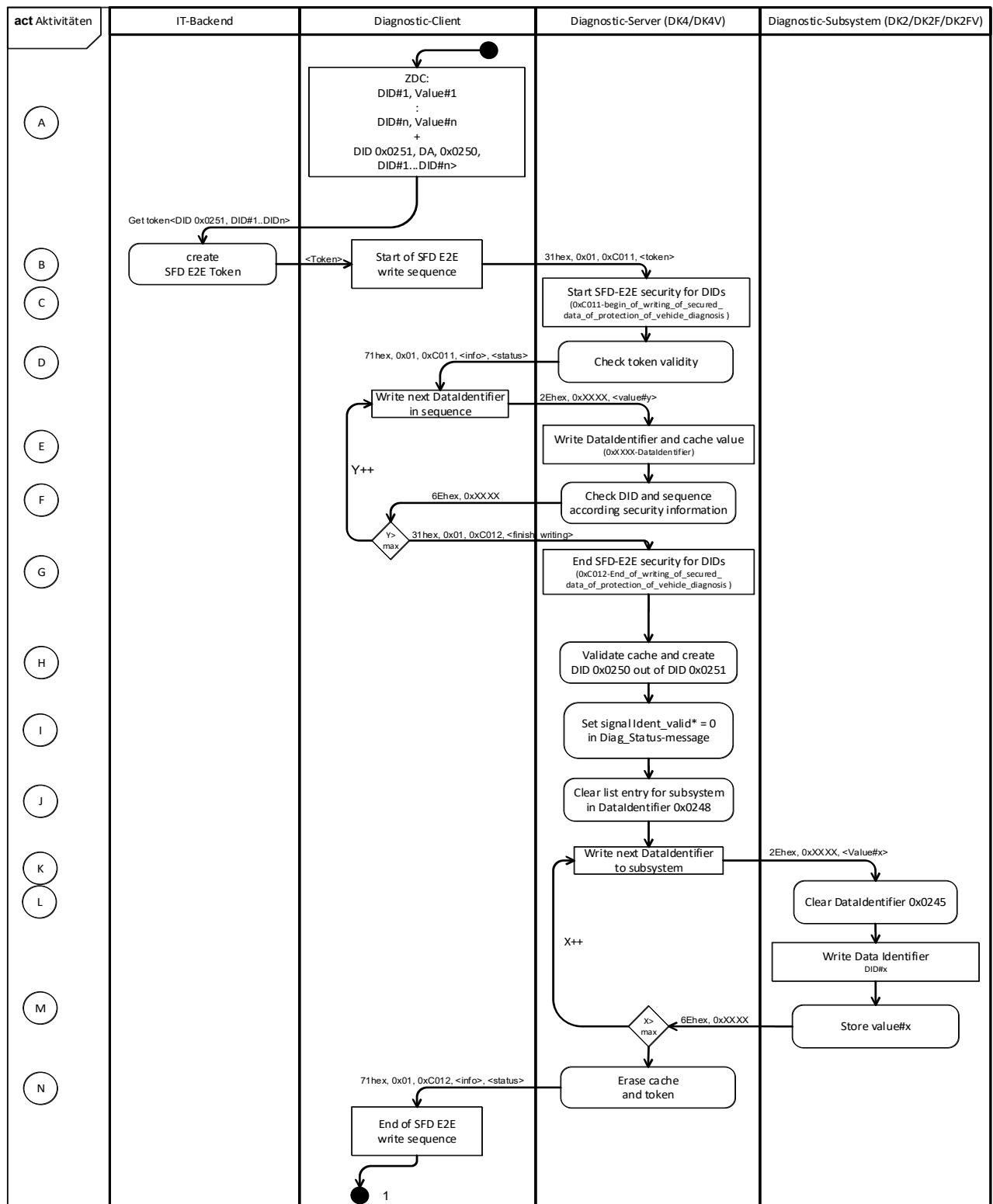
#### 3.1.1.1 Schreiben von SFD-Ende-zu-Ende abgesicherten Daten in ein DK2/DK2F/DK2FV-System

[I: F-LAH\_RxSWIN-280]

Ein Diagnose-Client schreibt SFD-Ende-zu-Ende abgesicherte Konfigurationsdaten über ein DK4-low/DK4V-low-System in ein unterlagertes DK2/DK2F/DK2FV-System.

[/: F-LAH\_RxSWIN-281]

**Abbildung 3-1 Schreiben von Anpassungen in ein DK2/DK2F/DK2FV-System, Teil 1/2**

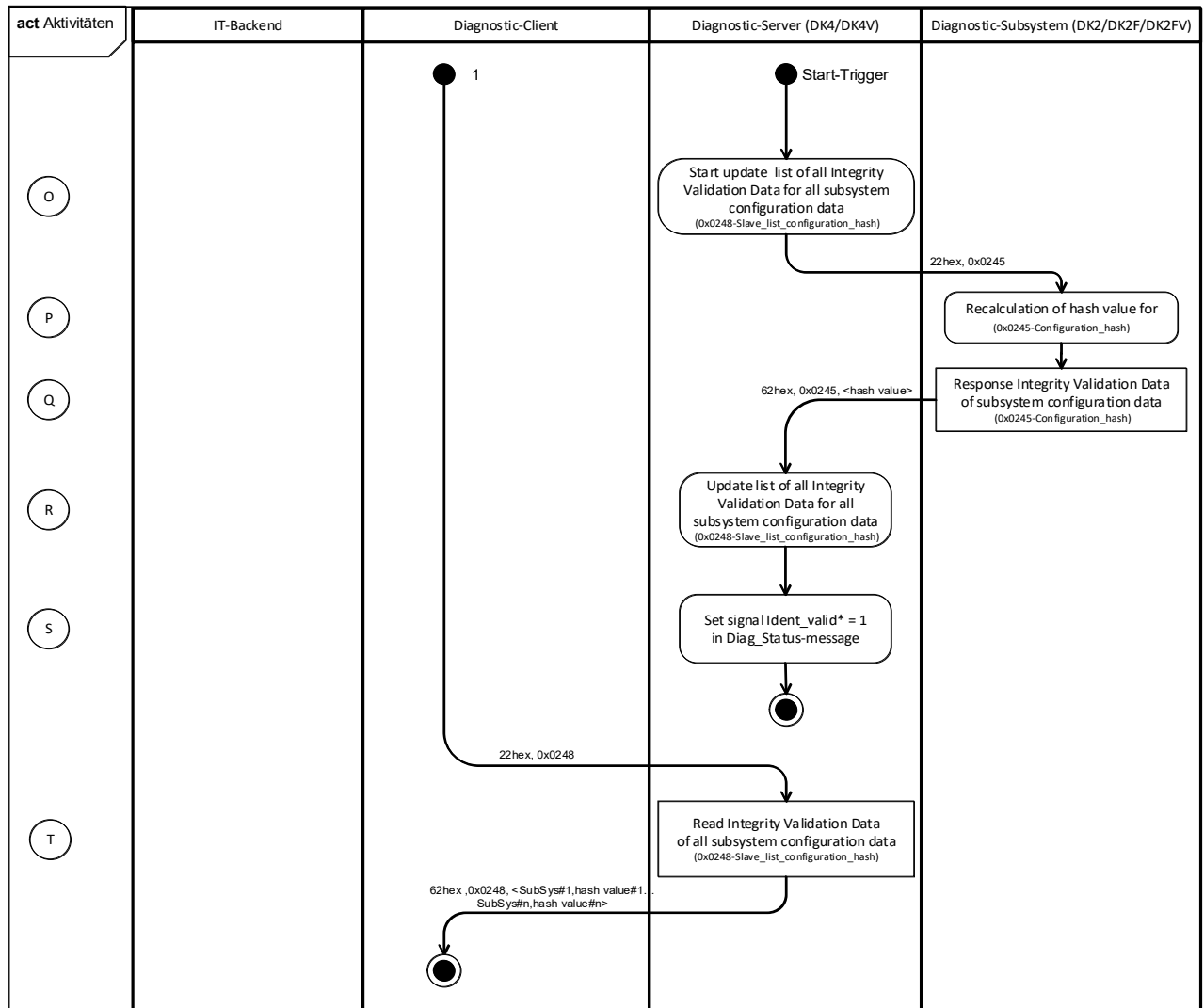


\*Note: The signal „Ident\_valid“ (QLAH 80114 from version 5.6) has only informational character here and no requirement character.



[/ F-LAH\_RxSWIN-282]

**Abbildung 3-2 Schreiben von Anpassungen in ein DK2/DK2F/DK2FV-System, Teil 2/2**



\*Note: The signal „Ident\_valid“ (QLAH 80114 from version 5.6) has only informational character here and no requirement character.

[I: F-LAH\_RxSWIN-1190]

Hinweis: Das Subsystem-Handling eines DK4 Steuergeräts von unterlagerten DK2(F) Steuergeräten, wie in den Schritten H - M beschrieben, ist nicht Bestandteil der IVD oder SFD Standard-Software und muss systemspezifisch umgesetzt werden.

[I: F-LAH\_RxSWIN-283]

A – Die für die Steuergeräte-Konfiguration notwendigen DataIdentifier werden dem Diagnose-Client zur Verfügung gestellt. Die DataIdentifier ergeben sich durch den Bauauftrag, der den ZDC konfiguriert. Hieraus resultiert auch der Inhalt des DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" für ein bestimmtes Subsystem. Bei SFD-E2E abgesicherte Diagnose-Systeme vor ZP8 fragt der Diagnostic-Client ein Absicherungs-Token am SFD-IT-Backend für den Diagnostic-Server an.

[I: F-LAH\_RxSWIN-284]

B\* - Das IT-Backend erstellt ein Absicherungs-Token über die DataIdentifier DID#1 ... DID#n und DID "0x0251-Write\_generic\_to\_sub\_system".

[I: F-LAH\_RxSWIN-285]

C\* - Der Absicherungs-Token wird mit dem Start der SFD-Routine "0xC011-Begin\_of\_writing\_of\_secured\_data\_of\_protection\_of\_vehicle\_diagnosis" an den Diagnostic-Server übertragen und leitet die Schreib-Sequenz, für die SFD-Ende-zu-Ende abgesicherte Daten, ein.

[I: F-LAH\_RxSWIN-286]

D\* - Der Diagnostic-Server prüft die Validität des empfangenen SFD Absicherungs-Tokens. Entsprechend der SFD-E2E Absicherung gemäß Dokument /4/ wird bei einem Empfang eines ungültigen Tokens die Absicherungsinformationen gelöscht.

[I: F-LAH\_RxSWIN-287]

E - Alle DataIdentifier, aus dem konfigurierten ZDC und der DataIdentifier "0x0251-Write\_generic\_to\_sub\_system" werden sequenziell mittels Service WriteDataByIdentifier (2Ehex) an den Diagnostic-Server übertragen und zwischengespeichert. Bei SFD mit Gruppenfreischaltung werden sie direkt vom Diagnostic-Server in die Diagnostic-Subsysteme geschrieben.

[I: F-LAH\_RxSWIN-288]

F\* - Entsprechend der SFD-E2E Absicherung nach Dokument /4/ wird die Empfangsreihenfolge der einzelnen DataIdentifier entsprechend der Absicherungsinformation überprüft. Bei positiver Prüfung antwortet der Diagnostic Server mit einer positiven Response. Bei einem negativen Prüfergebnis werden die empfangenen DataIdentifier und die zugehörigen DataRecords sowie die Absicherungsinformationen gelöscht. Bei einem negativen Prüfergebnis antwortet der Diagnostic-Server mit einem NRC "0x22-ConditionsNotCorrect".

[I: F-LAH\_RxSWIN-289]

G\* - Die Schreibsequenz des Diagnostic-Clients wird mit dem Start der Routine "0xC012-End\_of\_writing\_of\_secured\_data\_of\_protection\_of\_vehicle\_diagnosis" mit der RoutineControlOption [0x01-Finish\_writing\_of\_secured\_data] abgeschlossen.

[I: F-LAH\_RxSWIN-290]

H\* - Die Authentizität der übertragenen DataIdentifier wird anhand der Absicherungsinformation aus dem Absicherungs-Token geprüft. Die Adressierung des Ziel-Subsystems und der Inhalt des DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" wird anhand des empfangenen DataIdentifier "0x0251-Write\_generic\_to\_sub\_system" ermittelt.

[I: F-LAH\_RxSWIN-295]

I - In der Diag\_Status-Botschaft des Diagnostic-Server wird im Signal [Ident\_valid] mit dem Wert '0' angezeigt, dass die Identifikationsdaten aktuell nicht gültig sind. Durch das Speichern der empfangenen DataIdentifier ist der vorhanden Hashwert der Konfigurationsdaten für das Subsystem nicht mehr gültig und muss neu berechnet werden.

[I: F-LAH\_RxSWIN-716]

J - Der Listen-Eintrag im Sammel-Dataldentifizier "0x0248-Slave\_list\_configuration\_hash" für das Diagnostic Subsystem wird gelöscht.

[I: F-LAH\_RxSWIN-293]

K\* - Die zwischengespeicherten Dataldentifizier aus dem Zwischenspeicher werden sequentiell zu dem Diagnostic-Subsystem mittels Service WriteDataByIdentifizier (2Ehex) gesendet.

[I: F-LAH\_RxSWIN-717]

L - Mit dem Request zum Schreiben der Konfigurationsdaten wird der Dataldentifizier "0x0245-Configuration\_hash" im Diagnostic Subsystem wird gelöscht.

[I: F-LAH\_RxSWIN-291]

M\* - Das Diagnostic-Subsystem prüft die im Request empfangenen Daten auf Gültigkeit (z.B. Wertebereich). Bei einer positiven Prüfung werden die Daten im Subsystem gespeichert. Bei einer negativen Prüfung werden die Daten verworfen und ein negative Response zu dem Diagnostic-Server gesendet. Der Diagnostic-Server sendet einen Fehlercode im positiven Response der Routine "0xC012-End\_of\_writing\_of\_secured\_data\_of\_protection\_of\_vehicle\_diagnosis" zu dem Diagnostic-Client.

[I: F-LAH\_RxSWIN-296]

N\* - Der Zwischenspeicher und die Absicherungsinformationen in dem Diagnostic-Server werden gelöscht, nachdem alle empfangenen Dataldentifizier in den Zielspeicher des Diagnostic-Subsystem übertragen wurden.

[I: F-LAH\_RxSWIN-297]

O\* - Das Beenden der SFD-Schreibsequenz ist der "Start-Trigger" des Diagnostic-Server um den Sammel-Dataldentifizier „0x0248-Slave\_list\_configuration\_hash“ zu aktualisieren, indem er den Dataldentifizier "0x0245-Configuration\_hash" des Diagnostic-Subsystems ausliest.

[I: F-LAH\_RxSWIN-298]

P\* - Die Abfrage des Dataldentifizier "0x0245-Configuration\_hash" dient als Trigger zur Neuberechnung des Hashwertes der Konfigurationsdaten. Für die Neuberechnung werden genau die Dataldentifizier, in genau der Reihenfolge, aus der Liste, die mit "0x0250-Integrity\_validation\_data\_configuration\_list" übertragen wurden, einbezogen.

[I: F-LAH\_RxSWIN-299]

Q\* - Solange die Neuberechnung des Hashwertes noch nicht abgeschlossen ist, antwortet das Diagnostic-Subsystem mit einem NRC "0x78-RequestCorrectlyReceived-ResponsePending" dem Busmaster. Wenn die Neuberechnung abgeschlossen ist wird der Request mit einem positiven Response beantwortet.

[I: F-LAH\_RxSWIN-300]

R\* - Der Listeneintrag für das Diagnostic-Subsystem in dem Sammel-Dataldentifizier "0x0248 Slave\_list\_configuration\_hash" im Diagnostic-Server wird anhand des Dataldentifizier "0x0245-Configuration\_hash" aktualisiert.

[I: F-LAH\_RxSWIN-301]

S\* - In der Diag\_Status-Botschaft des Diagnostic-Server wird im Signal [Ident\_valid] mit dem Wert "1" angezeigt, dass die Identifikationsdaten vollständig gültig sind.

[I: F-LAH\_RxSWIN-695]

T - Der Diagnose-Client fragt den Sammel-Dataldentifizier "0x0248-Slave\_list\_configuration\_hash" für alle Diagnostic-Subsysteme ab.

[I: F-LAH\_RxSWIN-304]

\*) Nur relevant für SFD-E2E abgesicherte Diagnose-Systeme. Nicht relevant für Diagnose-Systeme mit Gruppenfreischaltung in der Produktion, vor ZP8.

### 3.2 ECU programming data security

[allg. Anf.: F-LAH\_RxSWIN-860]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Für neue Architekturen und die E<sup>3</sup> 1.2 - Architektur gilt:

[allg. Anf.: F-LAH\_RxSWIN-912]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Diagnose-Server müssen das Dokument /5/ (Q-LAH FDS) mindestens in der Version 3.1 oder ein alternatives, mindestens gleichwertiges Absicherungsverfahren, welches mit der E/E-Security-Abteilung des Auftraggebers abgestimmt ist, verwenden.

[allg. Anf.: F-LAH\_RxSWIN-862]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Diagnose-Server müssen das Dokument /6/ (Q-LAH 80126) mindestens in der Version 2.9 verwenden.

[allg. Anf.: F-LAH\_RxSWIN-864]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Diagnose-Server müssen das Dokument /14/ (Q-LAH 80128-3) mindestens in der Version 4.5 verwenden.

[allg. Anf.: F-LAH\_RxSWIN-913]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Für Bestandsarchitekturen gilt:

[allg. Anf.: F-LAH\_RxSWIN-914]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Der Auftragnehmer muss mit der E/E-Security- und der Diagnoseabteilung des Auftraggebers abstimmen (Dokumentenabgleich), welche Version des Dokuments /5/ (Q-LAH FDS) und somit welche Schlüssellänge oder welches alternative Absicherungsverfahren verwendet wird.

[allg. Anf.: F-LAH\_RxSWIN-915]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Der Auftragnehmer muss mit der E/E-Security- und der Diagnoseabteilung des Auftraggebers abstimmen (Dokumentenabgleich), welche Version des Dokuments /6/ (Q-LAH 80126) verwendet wird.

[allg. Anf.: F-LAH\_RxSWIN-916]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Der Auftragnehmer muss mit der E/E-Security- und der Diagnoseabteilung des Auftraggebers abstimmen (Dokumentenabgleich), welche Version des Dokuments /14/ (Q-LAH 80128-3) verwendet wird.

## 4 Integrity Validation Data

[I: F-LAH\_RxSWIN-786]

Integrity Validation Data werden verwendet:

[I: F-LAH\_RxSWIN-831]

- als Integritätsmerkmal von Software und Daten
- zur Überprüfung der Integrität von Steuergeräten in der Produktion
- zur Identifikation der Konfiguration von Steuergeräten per ZDC

[I: F-LAH\_RxSWIN-830]

[I: F-LAH\_RxSWIN-829]

### 4.1 Allgemeine Anforderungen

[I: F-LAH\_RxSWIN-225]

Für Instruction-Code (Applikations- und Bootloader-Software) und Konfigurationsdaten (Data-Identifizierer und Datensätze aus dem ZDC) werden separate Hashwerte berechnet. Die Berechnung der Hashwerte erfolgt sowohl in IT-Systemen und im Diagnose-Server. Dadurch ist ein Abgleich zwischen dem Sollzustand aus dem IT-Backend und dem Istzustand in den Diagnose-Servern möglich.

[I: F-LAH\_RxSWIN-233]

Jeder der Hashwerte wird vom Diagnose-Server als Integrity Validation Data zur Verfügung gestellt. Dies erfolgt je nach Diagnoseklasse per Data-Identifizierer oder Routinen-Identifizierer.

[allg. Anf.: F-LAH\_RxSWIN-1173]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Veränderbare Dateninhalte (Tool/Nutzer-Kennung Versionen von Linkern/Compilern etc., Zeitstempel, personenabhängige Signaturen/Zertifikate, Build-Versionen etc.), die im Erstellungsprozeß der Software entstehen und im Programmcode eingebunden werden, dürfen nicht in die Hashwert-Berechnung für die Integrity Validation Data (IVD) mit einbezogen werden.

Daher dürfen sie auch nicht in die CRC32 einfließen, die zur Hashwert-Berechnung verwendet wird.

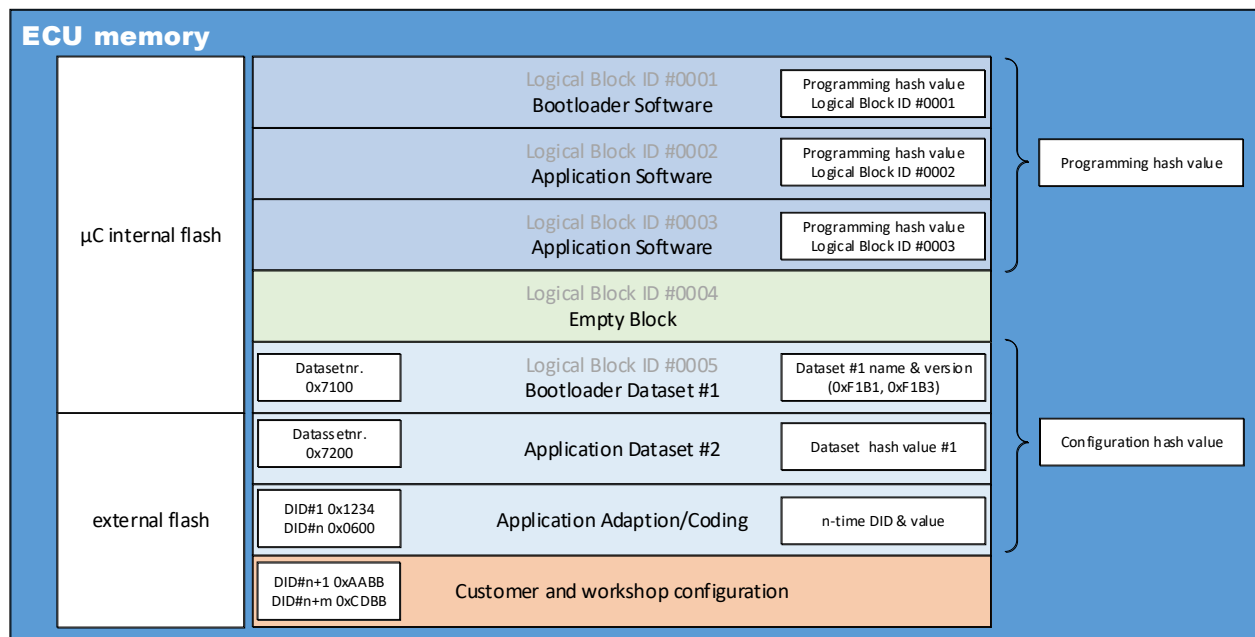
[allg. Anf.: F-LAH\_RxSWIN-1174]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Eine unveränderte Software (Instruction-Code) einer Software-Version darf durch unterschiedliche Build-Prozeduren nicht zu unterschiedlichen IVDs führen. Eine SW-Version hat immer dieselbe IVD. Nur Änderungen des Instruction-Codes dürfen zu einer geänderten IVD führen.

[I: F-LAH\_RxSWIN-234]

**Abbildung 4-1 Beispielhafte Steuergeräte-Speicherarchitektur**



[allg. Anf.: F-LAH\_RxSWIN-638]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Zur Berechnung des Programmierungs- und des Konfigurations-Hashwertes wird der SHA-256 Algorithmus verwendet.

[allg. Anf.: F-LAH\_RxSWIN-1002]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die zur Hashwertberechnung verwendeten

- logischen Werte, wie z.B. Data Identifier, Datensatznummern oder Checksummenwerte müssen als Hexadezimal-Byte-Array übernommen werden.
- symbolischen Werte bzw. Strings für die Identifikation, wie z.B. Software Version Nummer oder Datensatzname, müssen 7-Bit ASCII-Encoding verwenden, d.h. für jedes Zeichen wird 1 Byte im ASCII-Wertebereich verwendet. Bezüglich des Zeichenumfanges (Groß-, Kleinbuchstaben, Sonderzeichen und Ziffern) müssen die ASCII-Strings, wie in den Dokumenten /2/, /6/ und /14/ vorgegeben, umgesetzt werden.

[allg. Anf.: F-LAH\_RxSWIN-707]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Zeit zur Berechnung des Hashwertes darf zwischen Request und Response der Diagnose Routine "0x0253-Calculate\_integrity\_validation\_data" (bzw. Data Identifier "0x0247-Slave\_list\_programming\_hash" und "0x0248-Slave\_list\_configuration\_hash") maximal 800 ms betragen. Abweichungen von dieser Dauer müssen mit der Diagnosefachabteilung, so wie mit Produktion und Kundendienst abgestimmt werden und in den komponentenspezifischen Diagnoseanforderungen des BT-LAH dokumentiert werden.

Hinweis: Die Dauer der CRC32-Berechnung ist in den 800ms nicht mit enthalten, da für die Hashwert-Berechnung von vorberechneten CRC32 Checksummen ausgegangen wird.

[I: F-LAH\_RxSWIN-1018]

Die Berechnung des Hashwertes muss authentisch erfolgen. D.h. der Instruction-Code für die Routine zur Berechnung des Hashwertes darf nicht unberechtigt manipulierbar sein oder eine Manipulation des Codes muss nachvollziehbar sein. Solch ein Schutz kann zum Beispiel mittels EPDS und Secure Boot oder Installation in controllerinternem Speicher sichergestellt werden. Eine Abschwächung des Schutzniveaus ist nur in Rücksprache mit der Diagnosefachabteilung möglich.

[I: F-LAH\_RxSWIN-994]

Hinweis: Kann das Steuergerät aufgrund der Berechnung des Hashwertes nicht innerhalb von 50 ms antworten, ist mit einem NRC "0x78-RequestCorrectlyReceived-ResponsePending" zu antworten.

[I: F-LAH\_RxSWIN-940]

**Tabelle 4-1 Zuordnung von Datenkategorie (Datenart nach Dokument /8/) zu Hashwert**

|   | Programming hash | Configuration hash |
|---|------------------|--------------------|
| <b>Programmdaten</b><br>(Program data)                    | x                | NoImp              |
| <b>Applikationsdaten</b><br>(Calibration data)            | x                | NoImp              |
| <b>Codierung</b><br>(Coding)                              | NoImp            | x                  |
| <b>Fahrzeugparameter</b><br>(Vehicle parameters)          | NoImp            | x                  |
| <b>Erstbedatungswerte</b><br>(Initial calibration values) | NoImp            | x                  |
| <b>Kundenparameter</b><br>(Customer parameters)           | NoImp            | NoImp              |
| <b>Werkstattparameter</b><br>(Workshop parameters)        | NoImp            | NoImp              |
| <b>Prozessparameter</b><br>(Process parameters)           | NoImp            | NoImp              |
| <b>Lernwerte</b><br>(Learned values)                      | NoImp            | NoImp              |
| <b>Analysedaten</b><br>(Analysis data)                    | NoImp            | NoImp              |

[I: F-LAH\_RxSWIN-944]

x = beinhaltet

[I: F-LAH\_RxSWIN-981]

Für die standardisierten Identifikationsdaten nach Dokument /2/ und /12/ ist eine Zuordnung generischer Diagnoseobjekte zu Datenkategorien in der Diagnosefachabteilung zu erfragen.

#### 4.1.1 Programmierungs-Hashwert

[allg. Anf.: F-LAH\_RxSWIN-659]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die standardisierten Identifikationsdaten nach Dokument /2/ und /12/ sind bei der Berechnung des Programmierungs-Hashwertes unter den folgenden Bedingungen (UND-Verknüpfung) berücksichtigt, wenn sie:

[allg. Anf.: F-LAH\_RxSWIN-817]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- weder über den Service WriteDataByIdentifier (2Ehex) noch über den Datensatzdownload (Bootloader/Applikation) änderbar sind.

[allg. Anf.: F-LAH\_RxSWIN-816]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- nur über Programmierung änderbar sind.

[allg. Anf.: F-LAH\_RxSWIN-815]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- entsprechend der Diagnoseklasse gefordert sind.

[allg. Anf.: F-LAH\_RxSWIN-814]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- entsprechend der OBD-Klasse gefordert sind.

[allg. Anf.: F-LAH\_RxSWIN-813]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- entsprechend der Umsetzungsanforderungen für bestimmte Anwendungsfälle gefordert sind.

[allg. Anf.: F-LAH\_RxSWIN-812]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- entsprechend der Systemauslegung gem. Dokument /12/ gefordert sind.

[I: F-LAH\_RxSWIN-660]

Beispielhaft sind die folgenden DataIdentifier nach der Dokument /2/ und /12/ in Abhängigkeit der OBD-Relevanz, der Diagnoseklasse, der Umsetzungsanforderungen und der Systemauslegung in dem Programmierungs-Hashwert enthalten.

Nur über Programmierung änderbar (Diagnoseklassen DK2F und höher):

- 0xF187 - VW Spare Part Number
- 0xF189 - VW Application Software Version Number
- 0xF19E - ASAM ODX File Identifier
- 0xF1A2 - ASAM ODX File Version

Nur für Systeme, die Software Compositions enthalten:

- 0x0441 - SWCO\_list\_nr
- 0x011D - SWCO\_list\_system\_name

Nur für OBD-relevante Systeme:

- 0x02CE - OBD\_type
- 0x02CF - OBD\_class\_description



[/ : F-LAH\_RxSWIN-688]

Durch die Hashwertberechnung der Programmdateien werden folgende Datenkategorien (Datenarten) gemäß Dokument /8/ einbezogen:

[allg. Anf.: F-LAH\_RxSWIN-938]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Programmdateien (Programmcode oder Software für das Steuergerät)  
Hinweis: Programmdateien sind Bestandteil des Flashcontainers.

[allg. Anf.: F-LAH\_RxSWIN-939]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Applikationsdateien (ausstattungsabhängige Daten und Kennlinien)  
Hinweis: Applikationsdateien sind Bestandteil des Flashcontainers.

#### 4.1.1.1 Anforderungen an Embedded-Systeme

[allg. Anf.: F-LAH\_RxSWIN-423]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Anforderungen dieses Kapitels gelten für alle programmierbaren Diagnose-Server entsprechend Dokument /6/.

[allg. Anf.: F-LAH\_RxSWIN-238]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Integrity Validation Data für die Programmierung (Instruction-Code) umfasst grundsätzlich alle logischen Blöcke der Applikations- und Bootladersoftware.

[allg. Anf.: F-LAH\_RxSWIN-232]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Für alle logischen Blöcke der Applikations- und Bootladersoftware im Speicher eines Steuergerätes entsprechend Dokument /6/ wird ein einziger Hashwert durch das Steuergerät berechnet.

[allg. Anf.: F-LAH\_RxSWIN-740]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Bei Verwendung von segmentierten Blöcken entsprechend Dokument /6/ erfolgt die Berechnung der Integrity Validation Data immer über den gesamten logischen Block inklusive aller Segmente.

[allg. Anf.: F-LAH\_RxSWIN-570]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Als Bestandteil der Applikations- bzw. Bootlader Software werden auch die Metadaten (zur SW-Konfiguration z. B. für Adaptive Autosar oder Switch-Konfigurationen) und die Default-Daten und Default-Datensätze gezählt.

[Prozess-Anf.: F-LAH\_RxSWIN-1003]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Zur Hashwertberechnung werden nur Datenblöcke aus dem Flash-Container verwendet, die als Typ [FLASH\_DATA] gekennzeichnet sind.

[Prozess-Anf.: F-LAH\_RxSWIN-1179]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Zur Hashwertberechnung müssen immer alle updatebaren logischen Datenblöcke für die vollständige Applikations- und Bootlader-Software im dem Flash-Container, enthalten sein.

[allg. Anf.: F-LAH\_RxSWIN-389]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Leere logische Software Blöcke, die als Vorhalt dienen, dürfen nicht in die Hashwertberechnung einbezogen werden.

[allg. Anf.: F-LAH\_RxSWIN-745]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Flash-Treiber, als logischer Block im RAM, dürfen nicht in die Hashwertberechnung einbezogen werden.

[allg. Anf.: F-LAH\_RxSWIN-390]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

In die Hashwertberechnung durch die Routine "0x0253-Calculate\_integrity\_validation\_data", mit der ControlOption [Type\_of\_calculation] = 0x01, dürfen nur logische Blöcke einbezogen werden, die im DataIdentifier "0xF1AB-Logical Software Block Version" mit einer gültigen Software Versionsnummer gekennzeichnet sind.

[allg. Anf.: F-LAH\_RxSWIN-1008]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Abweichend vom Dokument /6/ und /14/ ist das partielle Flashen für alle updatebaren Steuergeräte (nach Dokument /6/) verpflichtend umzusetzen.

[allg. Anf.: F-LAH\_RxSWIN-692]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Feste, nicht durch die VOLKSWAGEN AG updatebare, Softwareanteile werden bei der Berechnung des Programmierungs-Hashwertes nicht berücksichtigt.

[allg. Anf.: F-LAH\_RxSWIN-1129]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Bei einer proprietären Programmierung (z. B. über Nadeladapter beim Lieferanten) muss die Routine 0x0202-Check Memory beim Lieferanten ausgeführt werden, damit die Checksummen-Berechnung über den Programmcode berechnet und die Checksumme im Steuergerät gespeichert wird. Wenn bestimmte Security-Maßnahmen, wie z. B. "SecureBoot" umgesetzt werden im SG, kann über den Nadeladapter auch die Checksumme beim Lieferanten mit programmiert werden.

Hinweis: Die Checksumme muss im SG vorhanden sein, da der Programming-Hashwert am ZP8 überprüft wird. Zu diesem Zeitpunkt würde eine Neuberechnung der Checksumme zu lange benötigen. Daher muss die Checksumme vorberechnet im SG vorliegen.

[allg. Anf.: F-LAH\_RxSWIN-1191]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Checksumme muss im SG vorhanden sein, da der Programming-Hashwert am ZP8 überprüft wird. Zu diesem Zeitpunkt würde eine Neuberechnung der Checksumme zu lange benötigen. Daher muss die Checksumme vorberechnet im SG vorliegen.

[/: F-LAH\_RxSWIN-956]

Bei einer Mehrfachablage der Software wird für die Berechnung des Programming-Hashwertes der redundant abgelegte Teil der Software nicht berücksichtigt.

[allg. Anf.: F-LAH\_RxSWIN-567]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Durch die Routine "0x0253-Calculate\_integrity\_validation\_data", mit der ControlOption [Type\_of\_calculation] = 0x01, wird die Neuberechnung des Programmierungs-Hashwertes auf dem Steuergerät durchgeführt.

[allg. Anf.: F-LAH\_RxSWIN-564]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Mit dem Starten der Routine "0xFF00-Erase Memory", die einen logischen Block mit Instruction-Code adressiert, muss die CRC32-Checksumme des adressierten logischen Blocks gelöscht werden.

[allg. Anf.: F-LAH\_RxSWIN-565]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Mit der positiven Prüfung, durch die Routine "0x0202-Check Memory", dass der logische Block fehlerfrei übertragen wurde muss die CRC32-Checksumme des logischen Blocks neu berechnet werden.

[allg. Anf.: F-LAH\_RxSWIN-718]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Mit dem positiven Response für die Routine "0x0202-Check Memory", die einen logischen Block mit Instruction-Code adressiert, wird die berechnete CRC32-Checksumme des logischen Blocks persistent gespeichert.

[allg. Anf.: F-LAH\_RxSWIN-636]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Routine "0x0544-Verify\_partial\_software\_checksum" entsprechend Dokument /6/ ist umzusetzen.

[allg. Anf.: F-LAH\_RxSWIN-617]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Mit dem Empfang des Requests für die Routine "0x0544-Verify\_partial\_software\_checksum" muss die Neuberechnung der CRC32-Checksumme für den im Request adressierten logischen Block durchgeführt und persistent im Steuergerät abgespeichert werden.

[allg. Anf.: F-LAH\_RxSWIN-1188]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Für Steuergeräte die CRC32 Checksummen für die Hashwertberechnung nicht auf Request der Routine "0x0544-Verify\_partial\_software\_checksum" Neuberechnen und speichern können ist es zulässig, dass in diesen Steuergeräten vorberechnete CRC32 Checksummen genutzt werden, wenn diese CRC32 Checksummen durch das Steuergerät mindestens einmal pro Fahrzyklus bei Startup automatisch neu berechnet, überprüft und gespeichert werden.

[allg. Anf.: F-LAH\_RxSWIN-345]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Berechnung des Programmierungs-Hashwertes im Steuergerät und IT Backend erfolgt über die konkatenierten Tupel, bestehend aus der logischen Block ID bzw. logischen Blocknummer (2 Bytes) mit der Byte-Reihenfolge big-endian, Software Version Nummer für jeden logischen Software-Block (4 Bytes ASCII-String) und CRC32-Checksumme jedes logischen Blocks (jeweils 4 Bytes) gemäß Dokument /6/ und wie im Abbildung 4-2 dargestellt.

[allg. Anf.: F-LAH\_RxSWIN-1184]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

In Flashcontainern entsprechend Dokument /14/ müssen die logischen Block IDs bzw. logischen Blocknummern wie folgt angegeben werden:

- 1 Byte Länge bei ALFID 0141
- oder
- 2 Byte Länge bei ALFID 0242.

[allg. Anf.: F-LAH\_RxSWIN-1185]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Steuergeräte, die logische Blöcke über eine 1 Byte lange Block-ID bzw. logischen Blocknummer adressieren, fügen in die Block-ID 1 Byte als führende Null ein.

[Prozess-Anf.: F-LAH\_RxSWIN-1186]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Wenn in Flashcontainern entsprechend Dokument /14/ die logischen Block IDs bzw. logischen Blocknummern mit 1 Byte Länge angegeben sind, wird im Backend 1 Byte als führende Null in die Block-ID eingefügt.

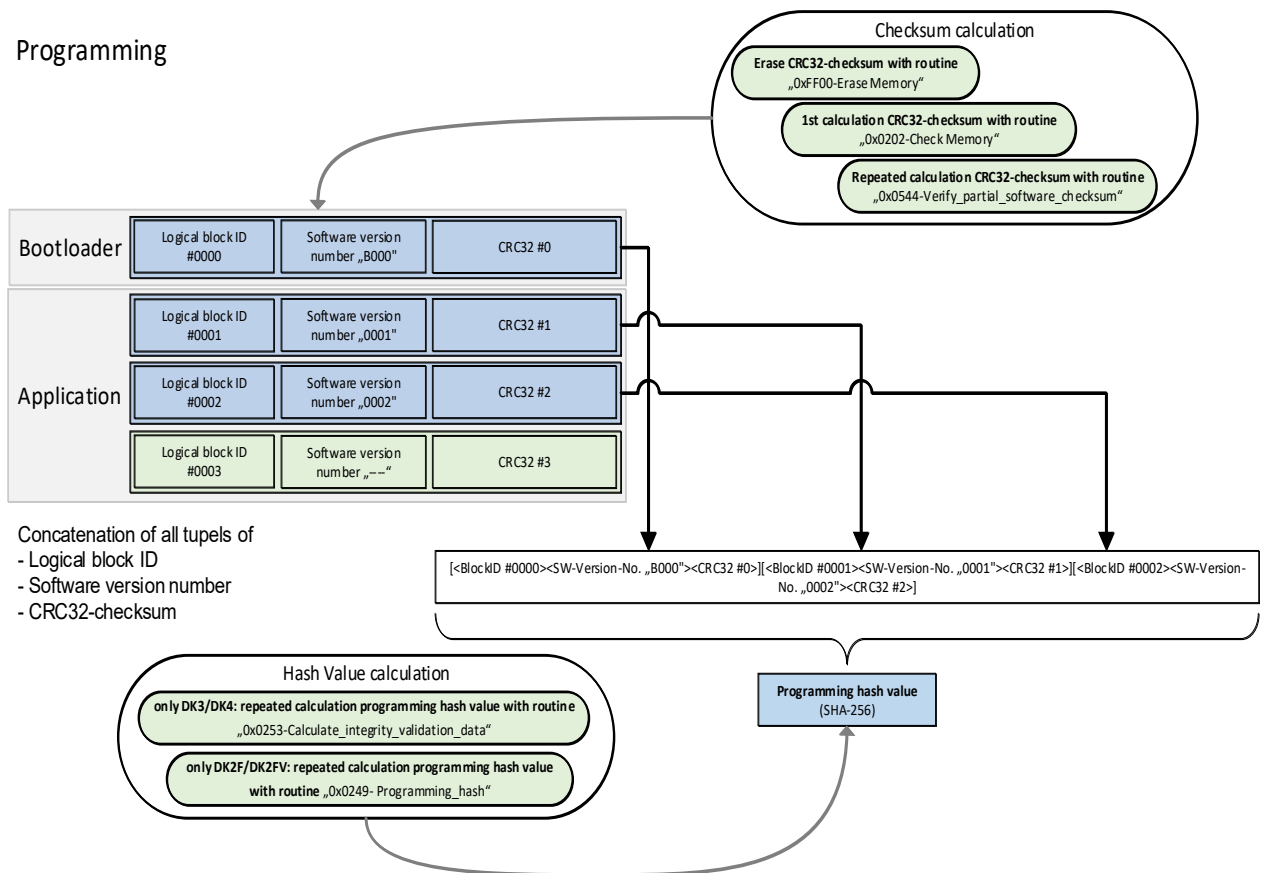
[allg. Anf.: F-LAH\_RxSWIN-715]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Tupel zur Berechnung des Programmierungs-Hashwertes werden in aufsteigender Reihenfolge der logischen Block ID konkateniert.

[I: F-LAH\_RxSWIN-235]

**Abbildung 4-2 Konkatination der Daten der logischen Blöcke der Applikation und des Bootloaders (Instruction-Code) zur Hashwertberechnung**



[I: F-LAH\_RxSWIN-388]

**Hinweis:**

- Logische Block ID (Länge = 2 Byte) gemäß Dokument /6/
- Programming-Hash-Value = Hashwert über den logischen Block der Applikations- bzw. der Bootloader-Software

[Prozess-Anf.: F-LAH\_RxSWIN-770]

[Bsm-E.D: nein, Bsm-E.L: nein, Bsm-E.T: nein, Bsm-O: nein, Bsm-Sa.FuSi: nein]

Die Berechnung des Soll-Programmierungs-Hashwertes erfolgt im IT-System analog der Berechnung im Diagnose-Server. Die CRC-32-Werte müssen dazu dem Flash-pdx entnommen werden.

#### 4.1.1.2 Anforderungen an nonEmbedded-Systeme bzw. filebasierte Systeme oder Embedded-Systeme mit systemspezifischer IVD-Berechnungsmethode

[! : F-LAH\_RxSWIN-1080]

Integrity Validation Data für die Programmierung (Instruction-Code) umfasst grundsätzlich alle Applikations- und Bootloadersoftware.

[! : F-LAH\_RxSWIN-1175]

Die Anforderungen dieses Kapitels gelten für alle programmierbaren Diagnose-Server entsprechend Dokument /6/, wo die CRC32-Checksumme der logischen Blöcke nicht ausschließlich von der Applikations- bzw. Bootloadersoftware, sondern auch von zusätzlichen Metadaten (z.B. Nutzerzertifikat, Zeitstempel, Logistikdaten, Randomzahl) abhängt.

[allg. Anf.: F-LAH\_RxSWIN-1176]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Integrity Validation Data für die Programmierung (Instruction-Code) darf ausschließlich eine Abhängigkeit zur Applikations- und Bootloadersoftware aufweisen.

[allg. Anf.: F-LAH\_RxSWIN-1177]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Integrity Validation Data für die Programmierung (Instruction-Code) darf keine Abhängigkeit zu Metadaten (z.B. Nutzerzertifikat, Zeitstempel, Logistikdaten, Randomzahl) der Software Blöcke der Applikations- und Bootloadersoftware aufweisen.

[allg. Anf.: F-LAH\_RxSWIN-1180]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Systeme (embedded oder non-embedded), die über eine systemspezifische IVD-Berechnungsmethode verfügen müssen die Routine „0x029A-Calculate\_module\_hash\_value“ umsetzen und dürfen die Routine „0x0544-Verify\_partial\_software\_checksum“ nicht umsetzen.

##### 4.1.1.2.1 Verwendung von Flashcontainer "ODX-Flash/PDX-Flash"

[allg. Anf.: F-LAH\_RxSWIN-1119]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Bei Verwendung von Flashcontainer entsprechend Dokument /14/ kann die Berechnung des Programmierungs-Hashwertes für nonEmbedded bzw. filebasierte Systeme oder Embedded-Systeme mit systemspezifischer IVD-Berechnungsmethode systemspezifisch erfolgen.

[allg. Anf.: F-LAH\_RxSWIN-1081]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Implementierungen der Berechnung des Programmierungs-Hashwertes ist mit dem Auftraggeber (Diagnosefachabteilung und Konzern-IT) abzustimmen.

[Prozess-Anf.: F-LAH\_RxSWIN-1082]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Im Falle von nonEmbedded-Systemen bzw. filebasierten Systemen oder Embedded-Systeme mit systemspezifischer IVD-Berechnungsmethode, die über UDS gemäß Dokument /6/ programmiert werden, muss der, vom Lieferanten im Offline-Prozess errechnete Soll-Hashwert der Programmierung und das verwendete Hash-Verfahren in der ODX-Flash-Datei dokumentiert und für IT-Systeme lesbar und auswertbar sein.

[Prozess-Anf.: F-LAH\_RxSWIN-641]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Alternativ kann bei Verwendung von Flashcontainer nach Dokument /14/ auch das Verfahren zur Berechnung des Programmierungs-Hashwertes entsprechend Kapitel "Anforderungen an Embedded-Systeme" angewendet werden. In diesem Fall darf das Unterelement „FW-CHECKSUM“ nicht verwendet werden.

#### 4.1.1.2.2 Verwendung von Flashcontainer abweichend von "ODX-Flash/PDX-Flash"

[allg. Anf.: F-LAH\_RxSWIN-643]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Bei Verwendung von Flashcontainer abweichend von Dokument /14/ kann die Berechnung system-spezifisch erfolgen.

[allg. Anf.: F-LAH\_RxSWIN-645]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Implementierungen der Hashwertberechnung ist mit dem Auftraggeber (Diagnosefachabteilung und Konzern-IT) abzustimmen.

[Prozess-Anf.: F-LAH\_RxSWIN-773]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Im Falle von nonEmbedded-Systemen bzw. filebasierten Systemen, die über einen Update Container gemäß Dokument /21/ programmiert werden, muss der vom Lieferanten im Offline-Prozess errechnete Soll-Hashwert der Programmierung, wie in Dokument /21/ (Datenelement „Identification-Data“) spezifiziert, über das sogenannte „Index file“ eines Update Containers dokumentiert und für IT-Systeme lesbar und auswertbar sein.

[Prozess-Anf.: F-LAH\_RxSWIN-644]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Da das Index-File für die Übertragung in die IT-Systeme nicht kryptographisch abgesichert ist, muss der vom Lieferanten im Offline-Prozess errechnete, Soll-Hashwert der Programmierung zusätzlich noch in einem kryptographisch abgesicherten Bereich des Update Containers, wie in Dokument /21/ spezifiziert, gespeichert werden.

[allg. Anf.: F-LAH\_RxSWIN-774]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Unabhängig von etwaigen im Fahrzeug verbauten Steuergerätevarianten, muss ein im Fahrzeug berechnetes Integritätsmerkmal für eine Steuergeräteversion immer den gleichen Hashwert liefern, der dem in den IT-Systemen hinterlegten Soll-Hashwert entspricht.

[allg. Anf.: F-LAH\_RxSWIN-775]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Beim Zugriff auf ein Integritätsmerkmal muss der Diagnose-Server die Integrität (Korrektheit) des Hashwerts des Merkmals sicherstellen. Sicherstellen der Integrität bedeutet in diesem Zusammenhang, dass bei einem unveränderten Hashwert auch alle dem Integrationsmerkmal zugeordneten Module unverändert geblieben sind. Der Algorithmus zur Berechnung eines Hashwertes eines Integrationsmarkmals aus Daten für einzelne Module ist mit dem Auftraggeber (Diagnosefachabteilung und Konzern-IT) abzustimmen.

[allg. Anf.: F-LAH\_RxSWIN-1088]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Im Falle von variantenbehafteten Modulen kann der Hashwert eines Moduls aus der Verrechnung der installierten Modulvariante und den nicht installierten Modulvarianten berechnet werden, wobei:

[allg. Anf.: F-LAH\_RxSWIN-1089]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- für installierte Modulvarianten ein Hashwert über die tatsächlich programmierten Daten berechnet wird und für die nicht installierte Modulvarianten ein eingebrachter Soll-Hashwert herangezogen wird.

[allg. Anf.: F-LAH\_RxSWIN-1090]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- für Module, die aufgrund der Steuergerätvariante nicht installiert wurden, kann der Hashwert ausschließlich aus eingebrachten Soll-Hashwerten, der nicht installierten Modulvarianten, berechnet werden.

#### 4.1.1.2.3 Neuberechnung des Programmierungs-Hashwert

[allg. Anf.: F-LAH\_RxSWIN-1092]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Für die Abfrage des Integritätsmerkmal mittels der Routinedentifizier „0x0253-Calculate\_integrity\_validation\_data“ auf einem DK3/DK3V/DK4/DK4V System oder der mittels DataIdentifizier „0x0249-Programming\_hash“ auf einem DK2F/DK2FV System, mit einer Timing Anforderung von 800 ms gemäß [allg. Anf.: F-LAH\_RxSWIN-707], können folgende Verfahren zur Bestimmung von Eingangsgrößen (Hashwerten von Teilmodulen) zur Hashwert-Berechnung genutzt werden:

[allg. Anf.: F-LAH\_RxSWIN-1093]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Berechnung durch Hashwertbildung über programmierte Datablöcke (Software)

[allg. Anf.: F-LAH\_RxSWIN-1094]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Nutzung von vorberechneten Hashwerten, sofern die Integrität über kryptographische Mittel sichergestellt ist und der Hashwert zur Laufzeit beim Zugriff/Laden der programmierten Datenblöcke überprüft wird (z.B. dm-verity mit Überprüfung des Root-Hashes durch einen Secure Boot Mechanismus).

[allg. Anf.: F-LAH\_RxSWIN-1095]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Nutzung von vorberechneten Hashwerten, wenn der Hashwert durch den Server mindestens einmal pro Fahrzyklus automatisch neu berechnet bzw. überprüft wird.

##### 4.1.1.2.3.1 DK3/DK4/DK3V/DK4V-Systeme

[allg. Anf.: F-LAH\_RxSWIN-779]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Durch die Routine "0x0253-Calculate\_integrity\_validation\_data", mit der RoutineControlOption [Type\_of\_calculation] = 0x01, wird die Neuberechnung des Programmierungs-Hashwertes auf dem Diagnose-Server durchgeführt. Der Diagnose-Server soll hierbei folgende Fälle unterscheiden:

[allg. Anf.: F-LAH\_RxSWIN-780]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Der Hashwert konnte noch nicht aus den abgesicherten Daten berechnet oder gegen die Daten geprüft werden.  
In diesem Fall darf der Diagnose-Server keinen Programmierungs-Hashwert zurück liefern. Im Parameter [Result\_of\_calculation] ist in diesem Fall der Wert "0x04 = Calculation\_incomplete" zurückzugeben.

[allg. Anf.: F-LAH\_RxSWIN-781]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Es existiert zur Laufzeit ein Prüfergebnis und der Diagnose-Server hat die Prüfung aller zu prüfenden Pakete (Module) abgeschlossen.  
In diesem Fall gibt der Diagnose-Server den errechneten Gesamthashwert der Programmierung, über alle dem Integrationsmerkmal zugeordneten Module zurück, unabhängig ob er dem Sollwert entspricht oder nicht. Im Parameter [Result\_of\_calculation] ist in diesem Fall der Wert "0x00 = Calculation\_successful" zurückzugeben.

[allg. Anf.: F-LAH\_RxSWIN-782]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Kann zur Laufzeit kein Prüfergebnis bzw. kein Hashwert ermittelt werden, weil dies in dem aktuellen Systemzustand verhindert wird (z.B. aktive Regeleingriffe), muss im Parameter [Result\_of\_calculation] der Wert „0x01 = Calculation\_failed“ zurückgegeben werden.

#### 4.1.1.2.4 DK2F/DK2FV-Systeme

[allg. Anf.: F-LAH\_RxSWIN-1099]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Durch den DataIdentifier "0x0249-Programming\_hash" wird die Neuberechnung des Programmierungs-Hashwertes auf dem Diagnose-Server durchgeführt. Der Diagnose-Server soll hierbei folgende Fälle unterscheiden:

[allg. Anf.: F-LAH\_RxSWIN-1100]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Der Hashwert konnte noch nicht aus den abgesicherten Daten berechnet oder gegen die Daten geprüft werden.  
In diesem Fall darf der Diagnose-Server keinen Programmierungs-Hashwert zurückgeben. Der Request ReadDataByIdentifier (22hex) mit dem DataIdentifier "0x0249-Programming\_hash" ist mit einem NRC "0x78-RequestCorrectlyReceived-ResponsePending" zu beantworten.

[allg. Anf.: F-LAH\_RxSWIN-1101]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Es existiert zur Laufzeit ein Prüfergebnis und der Diagnose-Server hat die Prüfung aller zu prüfenden Pakete (Module) abgeschlossen.  
In diesem Fall gibt der Diagnose-Server den errechneten Gesamthashwert der Programmierung über alle dem Integrationsmerkmal zugeordneten Module, in der positiven Response des DataIdentifier "0x0249-Programming\_hash" zurück, unabhängig ob er dem Sollwert entspricht oder nicht.

[allg. Anf.: F-LAH\_RxSWIN-1102]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Kann zur Laufzeit kein Prüfergebnis bzw. kein Hashwert ermittelt werden, weil dies in dem aktuellen Systemzustand verhindert wird (z.B. aktive Regeleingriffe), muss der Request ReadDataByIdentifier (22hex) mit dem DID "0x0249-Programming\_hash" ist mit einem NRC "0x22-ConditionNotCorrect" beantworten werden.



#### 4.1.1.2.5 Vollständige Neuberechnung des Programmierungs-Hashwert

[allg. Anf.: F-LAH\_RxSWIN-1104]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Für DK2F/DK2FV/DK3/DK3V/DK4/DK4V/SWCL-Systeme gilt:

- Die vollständige Neuberechnung aller Hashwerte systemspezifischer CRC32-Checksummen, die für die Berechnung des Integritätsmerkmals (Programmierung-Hashwert) benötigt werden, wird durch die Routine „0x029A-Calculate\_module\_hash\_value“ durchgeführt.

[allg. Anf.: F-LAH\_RxSWIN-1106]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Routine „0x029A-Calculate\_module\_hash\_value“ muss die Timing Anforderungen von 800 ms (gemäß [allg. Anf.: F-LAH\_RxSWIN-707]) bis zu einem positiven Response nicht erfüllen.

[allg. Anf.: F-LAH\_RxSWIN-1107]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Auf DK2F/DK2FV/DK3/DK3V/DK4/DK4V/SWCL-Systemen wird durch die Routine "0x029A-Calculate\_module\_hash\_value" mit der RoutineControlOption [Type\_of\_calculation] = 0x01 die vollständige Neuberechnung des Programmierungs-Hashwertes auf dem Diagnose-Server durchgeführt. Der Diagnose-Server soll hierbei folgende Fälle unterscheiden:

[allg. Anf.: F-LAH\_RxSWIN-1108]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Die vollständige Neuberechnung des Hashwertes konnte noch nicht berechnet werden. In diesem Fall darf der Diagnose-Server keinen Programmierungs-Hashwert zurück liefern. Im Response-Parameter [Result\_of\_calculation] ist in diesem Fall der Wert "0x04 = Calculation\_incomplete" zurückgeben.

[allg. Anf.: F-LAH\_RxSWIN-1109]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Die vollständige Neuberechnung des Hashwert konnte abgeschlossen werden. In diesem Fall gibt der Diagnose-Server den errechneten Gesamthashwert der Programmierung, über alle dem Integrationsmerkmal zugeordneten Module zurück, unabhängig ob er dem Sollwert entspricht oder nicht. Im Parameter [Result\_of\_calculation] ist in diesem Fall der Wert "0x00 = Calculation\_successful" zurückgeben.

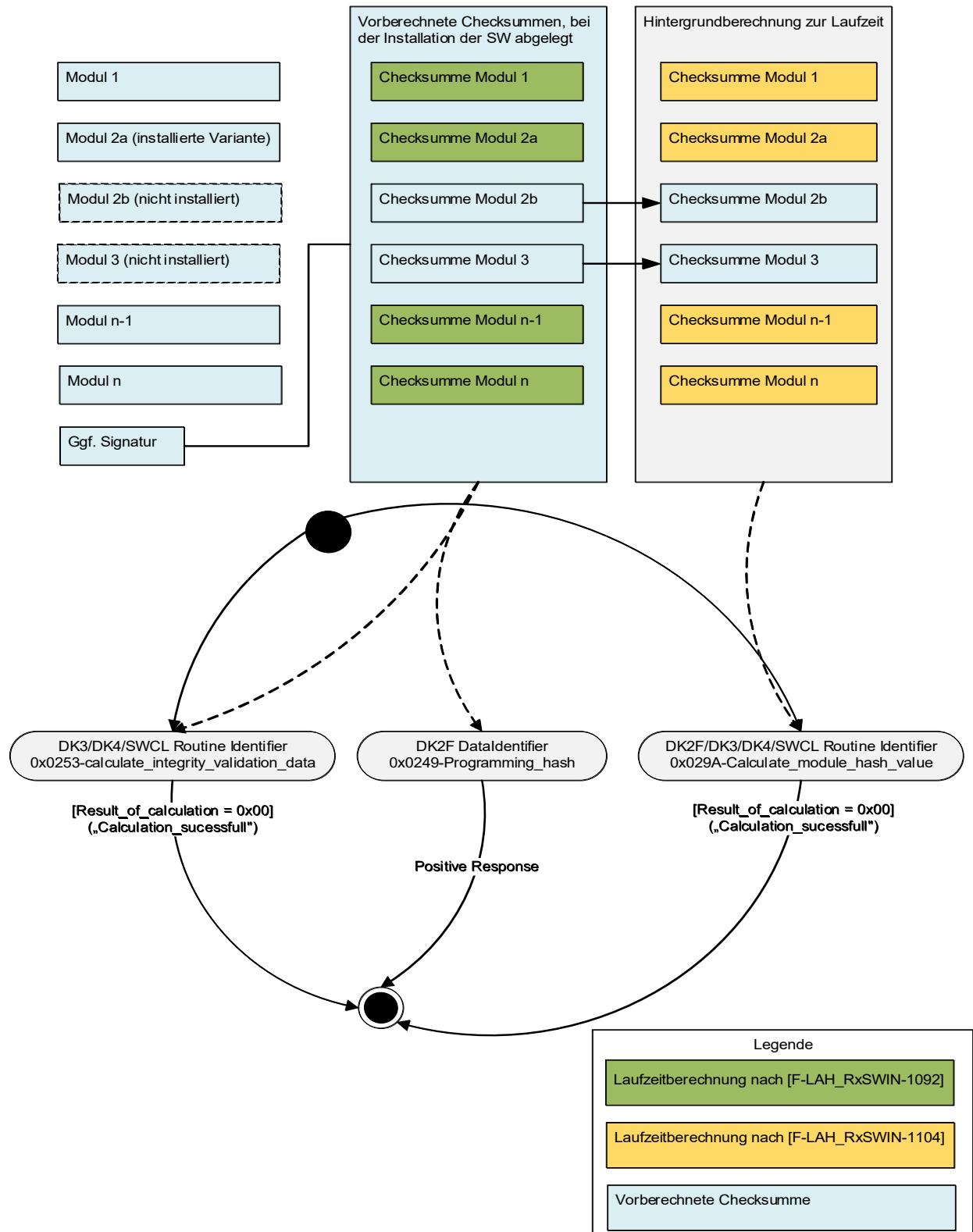
[allg. Anf.: F-LAH\_RxSWIN-1110]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Konnte die vollständige Neuberechnung des Hashwert nicht durchgeführt werden, weil dies in dem in dem aktuellen Systemzustand (z.B. aktive Regeleingriffe) verhindert wird, dann muss im Parameter [Result\_of\_calculation] der Wert „0x01 = Calculation\_failed“ zurückgegeben werden.

[I: F-LAH\_RxSWIN-924]

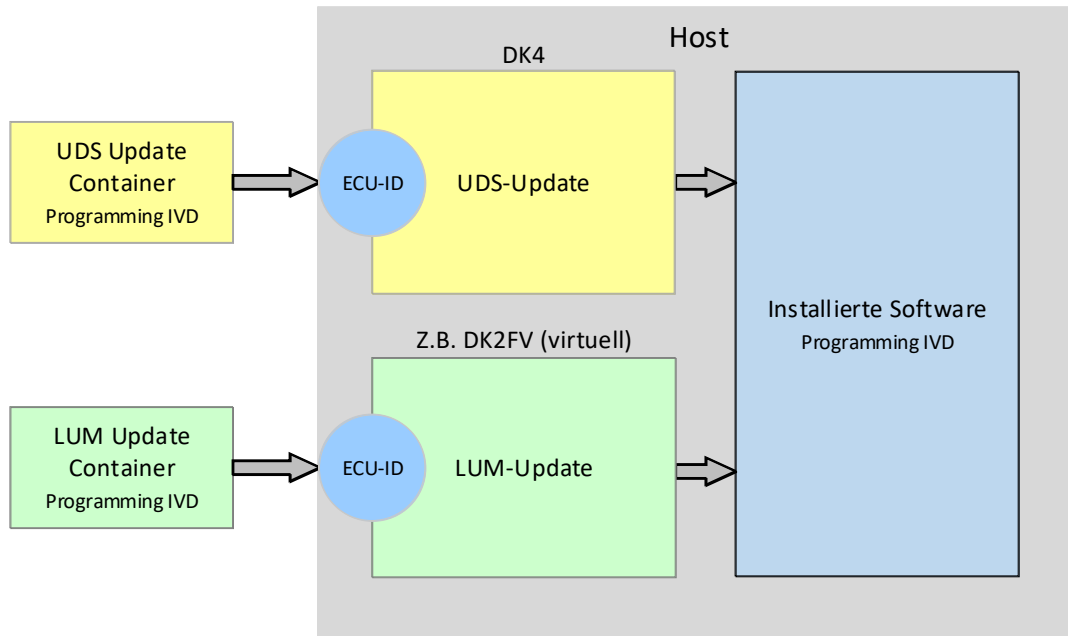
**Abbildung 4-3 Hashwertberechnung für LUM-Flashcontainer**



#### 4.1.1.2.6 Sonderfall UDS- und LUM-Flashen in einem Steuergerät

[I: F-LAH\_RxSWIN-1075]

**Abbildung 4-4 Exemplarische Darstellung eines Sonderfalls für Steuergeräte mit UDS- und LUM-Update**



[allg. Anf.: F-LAH\_RxSWIN-957]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Setzt ein Steuergerät sowohl eine UDS-Programmierung nach Dokument /6/ als auch das Verfahren mittels "Local Update Manager" (LUM) um, dann gilt folgendes:

[allg. Anf.: F-LAH\_RxSWIN-1130]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Der Programmierungs-Hashwert muss für beide Verfahren immer den selben Wert enthalten.
- [allg. Anf.: F-LAH\_RxSWIN-959]
- [BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]
- Der Programmierungs-Hashwert wird immer über beide Diagnoseadressen ausgegeben, unabhängig davon über welche Diagnoseadresse und mit welchem Verfahren das letzte Update durchgeführt wurde.

[allg. Anf.: F-LAH\_RxSWIN-960]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Konfiguration mittels ZDC darf nur über die Diagnoseadresse möglich sein, die eine Programmierung nach Dokument /6/ unterstützt.

[allg. Anf.: F-LAH\_RxSWIN-1069]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Partielle Softwareupdates bzw. Teile des Updates eines Steuergerätes auf mehrere Diagnoseadressen aufzuteilen ist nicht zulässig. Über jede der beiden Diagnoseadressen muss jeweils ein vollständiges Softwareupdate mit dem jeweiligen Verfahren durchgeführt werden.

[allg. Anf.: F-LAH\_RxSWIN-1070]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

In einem über zwei Diagnoseadressen programmierbaren Server muss unter allen Umständen nach einem Softwareupdate und unabhängig vom Programmierverfahren zum Zeitpunkt des Wiederaufstartens des Systems in den beiden Diagnoseadressen die aktuellen Identifikationsdaten (Logistikdaten entsprechend Dokument /6/) zur Verfügung stehen.

[allg. Anf.: F-LAH\_RxSWIN-1071]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Ein über zwei Diagnoseadressen programmierbarer Server muss dieselbe Information für die Identifikationsdaten aus physikalisch identischen Speicheradressen unabhängig vom aktuellen Betriebsmodus des Servers (Applikation oder Bootloader) und unabhängig von der Diagnoseadresse liefern.

[allg. Anf.: F-LAH\_RxSWIN-1072]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Ein über zwei Diagnoseadressen programmierbarer Server muss unter allen Umständen die Identifikationsdaten (Logistikdaten entsprechend Dokument /6/) der zuletzt "gültig" programmierten Software aus einem separaten Backup (nichtflüchtiger Speicher) in beiden Diagnoseadressen liefern können.

[allg. Anf.: F-LAH\_RxSWIN-1073]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Ausführung konkurrierender Diagnose-Services, z.B. Schreibzugriffe über 2Ehex-WriteData-Byldentifer oder konkurrierende Abläufe wie z.B. Softwareupdates, gleichzeitig über beide Diagnoseadressen muss durch entsprechende Maßnahmen verhindert werden.

Hinweis: Ziel ist es Abläufe, insbesondere für das Softwareupdate, nicht durch Diagnose-Services der jeweils anderen Diagnoseadresse zu beeinflussen.

[allg. Anf.: F-LAH\_RxSWIN-1074]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die nicht ausführbaren konkurrierenden Diagnose-Services müssen mit einem NRC „0x22-ConditionNotCorrect“ abgelehnt werden.

## 4.1.2 Konfigurations-Hashwert

[allg. Anf.: F-LAH\_RxSWIN-424]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Anforderungen dieses Kapitel gelten für alle konfigurierbaren Diagnoseserver.

[I: F-LAH\_RxSWIN-646]

Es wird nicht zwischen embedded- und filebasierten Systemen unterschieden.

[allg. Anf.: F-LAH\_RxSWIN-239]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der Hashwert für Konfigurationsdaten umfasst logische Blöcke mit Bootloader-Datensätzen und Konfigurationsdaten der Applikation.

[I: F-LAH\_RxSWIN-1016]

Die Reihenfolge der DataIdentifier und Datensatznummern für die Berechnung des Konfigurations-Hashwertes (IVD) ist unabhängig von der Reihenfolge, die bei der SFD-Ende-zu-Ende-Absicherung der Konfigurationsdaten verwendet wird.

[allg. Anf.: F-LAH\_RxSWIN-241]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

In die Hashwertberechnung der Konfigurationsdaten dürfen nur die folgenden Datenkategorien (Datenarten) gemäß Dokument /8/ einbezogen werden:

[allg. Anf.: F-LAH\_RxSWIN-791]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Codierung (ausstattungsabhängiges Ein/Ausschalten von Teilfunktionen)

[allg. Anf.: F-LAH\_RxSWIN-790]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Fahrzeugparameter (fahrzeugabhängige Parameter und Einstellung)

[allg. Anf.: F-LAH\_RxSWIN-789]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Erstbedatungswerte (einmalig (initial) beschreibbare Defaultwerte, Parameter und Einstellungen)

[allg. Anf.: F-LAH\_RxSWIN-308]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Nicht in die Hashwertberechnung der Applikations-Konfigurationsdaten dürfen die folgenden Datenkategorien (Datenarten) gemäß Dokument /8/ einbezogen werden:

[allg. Anf.: F-LAH\_RxSWIN-937]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Programmdatei (Programmcode oder Software für das Steuergerät)  
Hinweis: Programmdatei sind Bestandteil des Programmierungs-Hashwertes

[allg. Anf.: F-LAH\_RxSWIN-797]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Applikationsdaten (ausstattungsabhängige Daten und Kennlinien)  
Hinweis: Applikationsdaten sind Bestandteil des Programmierungs-Hashwertes

[allg. Anf.: F-LAH\_RxSWIN-796]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Kundenparameter (kundenabhängige Parameter und Einstellungen)

[allg. Anf.: F-LAH\_RxSWIN-795]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Werkstattparameter (werkstattabhängige Parameter und Einstellungen)  
Hinweis: Werkstattparameter dürfen nur innerhalb eines vorgegebenen Rahmens, ohne Beeinflussung der Typgenehmigung, verändert werden.

[allg. Anf.: F-LAH\_RxSWIN-794]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Prozessparameter (prozessabhängige Parameter und Einstellungen)  
Hinweis: Prozessparameter mit Einfluss auf die Typgenehmigung müssen in der Bauzustandsdokumentation dokumentiert werden.

[allg. Anf.: F-LAH\_RxSWIN-793]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Lernwerte (vom Steuergerät selbständig angelernte Parameter und Einstellungen)

[allg. Anf.: F-LAH\_RxSWIN-792]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Analysedaten (während des normalen Fahrzeugbetriebs dynamisch erzeugte Daten)

[allg. Anf.: F-LAH\_RxSWIN-785]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Nicht in die Hashwertberechnung der Applikations-Konfigurationsdaten darf der folgende Dataldentifizier einbezogen werden:

- 0xF18F-Regulation\_x\_software\_identification\_numbers

[Prozess-Anf.: F-LAH\_RxSWIN-787]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Hinweis: Der Dataldentifizier 0xF18F ist nicht Bestandteil des ZDC vom Gateway-Steuergerät und kann somit nicht bei der Berechnung des Soll-Wertes für den Konfigurations-Hashwert berücksichtigt werden. Er wird über einen separaten Datencontainer zugesteuert und vollständig in der Bauzustandsdokumentation (BZD) erfasst.

[allg. Anf.: F-LAH\_RxSWIN-544]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Eine Änderung von Kunden-, Werkstatt-, Prozessparametern, Lernwerten sowie Analysedaten darf NICHT zu einer Änderung der Integrity Validation Data von Konfigurationsdaten führen.

[allg. Anf.: F-LAH\_RxSWIN-568]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

"Vorgegebene Datensätze" gemäß Dokument /7/ gehen in die Hashwertberechnung der Konfigurationsdaten mit ein und müssen im ZDC referenziert sein.

[allg. Anf.: F-LAH\_RxSWIN-571]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

"Default Datensätze" gemäß Dokument /7/ zählen zur Applikations-Software und dürfen nicht in die Hashwertberechnung der Konfigurationsdaten einbezogen werden.

[allg. Anf.: F-LAH\_RxSWIN-772]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Reservierte Codierungen/Anpassungen gehen in die Hashwertberechnung der Konfigurationsdaten mit ein und müssen im ZDC enthalten sein.

[allg. Anf.: F-LAH\_RxSWIN-691]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Bei einer Mehrfachablage von z. B. ASIL-relevanten Daten wird für die Berechnung des Konfigurations-Hashwertes der redundant abgelegte Teil der Daten nicht berücksichtigt.

[allg. Anf.: F-LAH\_RxSWIN-647]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die standardisierten Identifikationsdaten nach Dokument /2/ und /12/ sind bei der Berechnung des Konfigurations-Hashwertes unter den folgenden Bedingungen (UND-Verknüpfung) zu berücksichtigen, wenn sie:

[allg. Anf.: F-LAH\_RxSWIN-822]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- über den Service WriteDataByIdentifier (2Ehex) oder über den Datensatzdownload (Bootloader/Applikation) änderbar sind.

[allg. Anf.: F-LAH\_RxSWIN-821]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- entsprechend der Diagnoseklasse gefordert sind.

[allg. Anf.: F-LAH\_RxSWIN-820]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- entsprechend der OBD-Klasse gefordert sind.

[allg. Anf.: F-LAH\_RxSWIN-819]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- entsprechend der Umsetzungsanforderungen für bestimmte Anwendungsfälle gefordert sind.

[allg. Anf.: F-LAH\_RxSWIN-818]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- entsprechend der Systemauslegung gem. Dokument /12/ gefordert sind.

[I: F-LAH\_RxSWIN-651]

Beispielhaft werden die folgenden DataIdentifier nach Dokument /2/ und /12/ in Abhängigkeit der Diagnoseklasse, der Umsetzungsanforderungen und der Systemauslegung dem Konfigurations-Hashwert zugeordnet.

Nur für Busmaster-Systeme der Diagnoseklasse 4-high:

- 0x04A3 - Gateway Component List
- 0x061A - Slave\_component\_list

Nur für Systeme, die den Datensatzdownload der Generation 2 unterstützen:

- 0xF1B1 - VW\_Application\_data\_set\_identification
- 0xF1B3 - VW\_Data\_set\_name

Nur für Systeme die Software Compositions enthalten:

- 0x0442 - SWCO\_list

[allg. Anf.: F-LAH\_RxSWIN-572]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Alle Konfigurationsdaten, die in die Hashwertberechnung der Konfigurationsdaten einbezogen werden, müssen Bestandteil des ZDCs sein.

[allg. Anf.: F-LAH\_RxSWIN-1017]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Konfigurationsdaten dürfen in einem Steuergerät nur über eine Methode (z.B. Anpassung, Datensatz) änderbar sein. Dies gilt auch für Steuergeräte mit integrierten Einheiten wie z.B. virtuellen Systemen oder Software Clustern.

[I: F-LAH\_RxSWIN-599]

Hinweis: Im ZDC müssen alle Konfigurationsdaten enthalten sein, damit im Feld durch den Kundendienst die Möglichkeit gegeben ist, alle Konfigurationsdaten entsprechend der Typgenehmigung wiederherstellen zu können, falls der Hashwert im Steuergerät nicht mit dem Soll-Hashwert im IT-System übereinstimmt.

[allg. Anf.: F-LAH\_RxSWIN-961]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Konfigurationsdaten eines Steuergerätes/Systems dürfen nicht im ZDC des über-/unterlagerten Systems enthalten sein. DK2-Systeme und DK4-Systeme müssen deshalb jeweils einen eigenen ZDC erhalten.

Hinweis: Dies ist notwendig, damit der Konfigurations-Hashwert im IT-System berechnet werden kann.

[allg. Anf.: F-LAH\_RxSWIN-623]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Daten der Wegfahrsperre (WFS), des Komponentenschutzes (KS), des Vehicle Key Management Systems (VKMS) und Daten von Software als Produkt (SWaP) bzw. Function on Demand (FOD) dürfen nicht in die Hashwertberechnung der Konfigurationsdaten einbezogen werden.

[/ F-LAH\_RxSWIN-624]

Hinweis: Die Daten der Wegfahrsperre (WFS) entsprechend Dokument /18/, des Komponentenschutzes (KS) entsprechend Dokument /19/, des Vehicle Key Management Systems (VKMS) entsprechend Dokument /15/ und Daten von Software als Produkt (SWaP) entsprechend Dokument /16/ bzw. Function on Demand (FOD) entsprechend Dokument /17/ werden abgesichert in das Fahrzeug eingebracht und in geschützten Speicherbereichen (SHE bzw. HSM) entsprechend Dokument /20/ abgelegt, so dass eine Manipulation der Daten ausgeschlossen werden kann.

[allg. Anf.: F-LAH\_RxSWIN-346]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Über den DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" wird die Liste mit DataIdentifiern für Anpassung/Codierung sowie Datensatznummern vorgegeben, die für die Hashwertberechnung über die Konfigurationsdaten verwendet werden soll.

[allg. Anf.: F-LAH\_RxSWIN-377]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Reihenfolge der Daten für die Hashwertberechnung ergibt sich aus dem Inhalt des DataIdentifiers "0x0250-Integrity\_validation\_data\_configuration\_list" und muss unverändert übernommen werden.



#### 4.1.2.1 Berechnung der Einzel-Hashwerte bzw. Checksummen der Konfigurationsdaten

[I: F-LAH\_RxSWIN-690]

Der Hashwert der Konfigurationsdaten wird mit Hilfe von verschiedenen einzelnen Hashwerten ermittelt. Diese Einzel-Hashwerte können über den Routinidentifizier "0x0254-Calculate\_individual\_hash\_value" ausgelesen werden.

##### 4.1.2.1.1 Einzel-Hashwerte für Anpassungen/Codierung und Applikations-Datensätze gemäß Dokument 7/

[allg. Anf.: F-LAH\_RxSWIN-593]

[Bsm-E.D: nein, Bsm-E.L: nein, Bsm-E.T: nein, Bsm-O: nein, Bsm-Sa.FuSi: nein]

Alle Steuergeräte berechnen den Einzel-Hashwert für alle Anpassungen/Codierung und die Einzel-Hashwerte für Applikations-Datensätze gemäß Dokument 7/ mit dem Empfang des Requests für die Routine "0x0253-Calculate\_integrity\_validation\_data" mit der RoutineControlOption [Type\_of\_calculation] = 0x00.

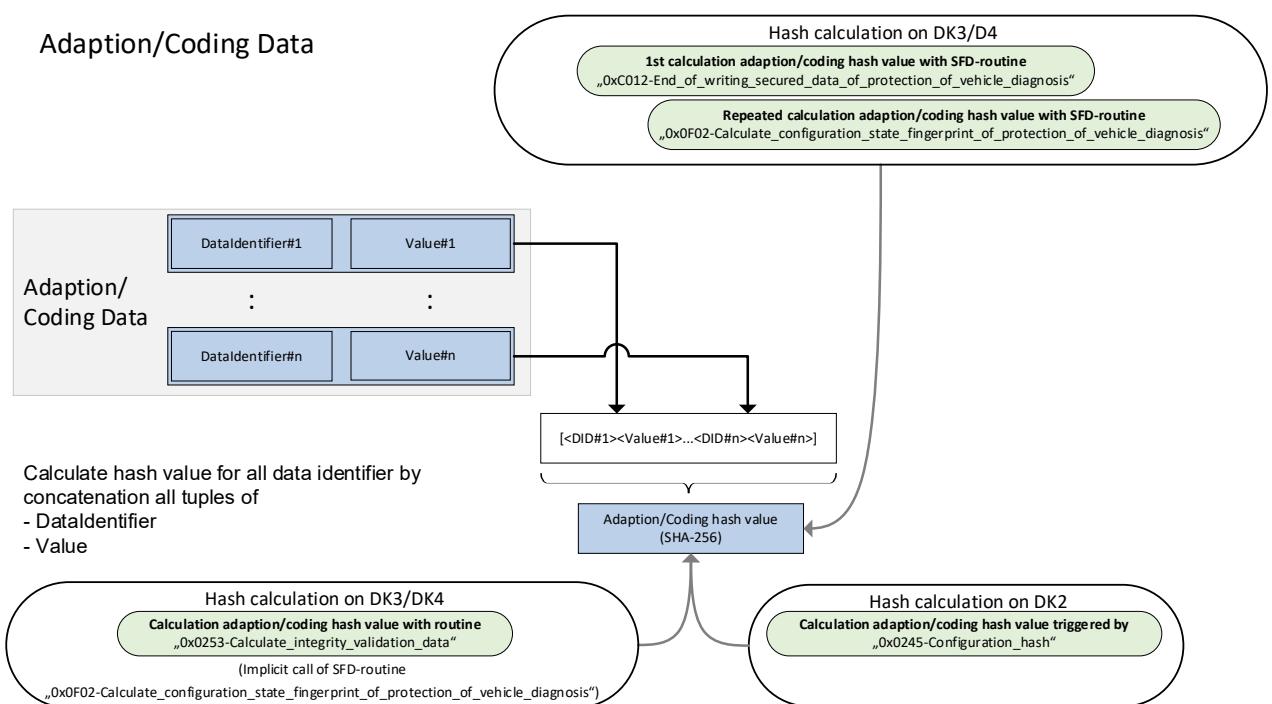
[allg. Anf.: F-LAH\_RxSWIN-1004]

[Bsm-E.D: nein, Bsm-E.L: nein, Bsm-E.T: nein, Bsm-O: nein, Bsm-Sa.FuSi: nein]

Der Einzel-Hashwert für Anpassungen/Codierung wird aus der Konkatenation aller Tupel, bestehend aus DataIdentifier gefolgt von dem jeweiligen DataRecord (Werte bzw. Byte-Arrays), berechnet.

[I: F-LAH\_RxSWIN-578]

#### Abbildung 4-4 Einzel-Hashwertberechnung über alle Anpassungen und Codierungen



[I: F-LAH\_RxSWIN-995]

Dieser Ablauf gilt auch für die Berechnung und das Auslesen von Einzel-Hashwerten mittels der Routine "0x0254- Calculate\_individual\_hash\_value".

[allg. Anf.: F-LAH\_RxSWIN-1005]

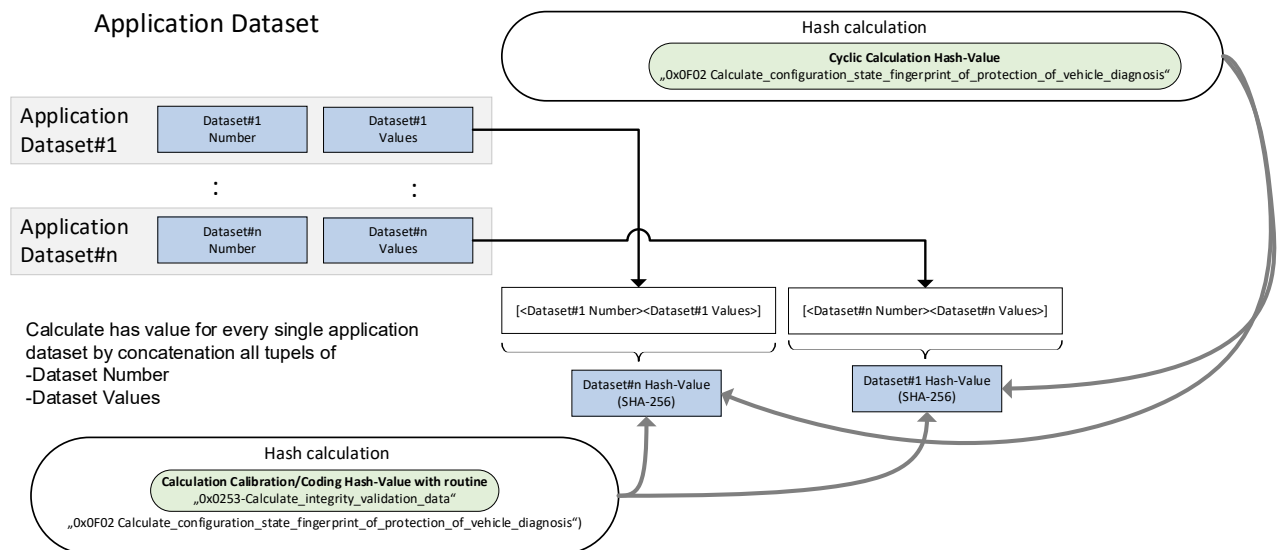
[Bsm-E.D: nein, Bsm-E.L: nein, Bsm-E.T: nein, Bsm-O: nein, Bsm-Sa.FuSi: nein]

Der Einzel-Hashwert für einen Applikationsdatensatz wird über das konkatenierte Tupel, bestehend aus der Datensatznummer und dem jeweiligen DataRecord (Werte bzw. Byte-Array), berechnet.

[I: F-LAH\_RxSWIN-585]

## Abbildung 4-5 Einzel-Hashwertberechnung über Applikations-Datensätze

Dataset Download Generation 2  
Application Dataset



[I: F-LAH\_RxSWIN-996]

Dieser Ablauf gilt auch für die Berechnung und das Auslesen von Einzel-Hashwerten mittels der Routine "0x0254- Calculate\_individual\_hash\_value".

[I: F-LAH\_RxSWIN-1000]

Unter Dataset #n Values werden alle Programmierdaten aus dem Datensatzcontainer zwischen den Tags <DATEN>...</DATEN> verstanden.

#### 4.1.2.1.2 CRC32-Checksummenberechnung für Bootloader Datensätze der Datensatzdownload Generation 2 gemäß Dokument /7/

[allg. Anf.: F-LAH\_RxSWIN-587]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die CRC32-Checksumme eines Bootloader-Datensatzes wird mit dem Empfang des Requests für die Routine "0xFF00-Erase Memory", die den Bootloader-Datensatz adressiert, gelöscht.

[allg. Anf.: F-LAH\_RxSWIN-714]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Routine "0xFF00-Erase Memory", die einen logischen Block mit Bootloader-Datensatz adressiert, darf den Programmierungs-Hashwert nicht beeinflussen.

[allg. Anf.: F-LAH\_RxSWIN-588]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die CRC32-Checksumme eines Bootloader-Datensatzes wird mit dem Empfang des Requests für die Routine "0x0202-Check Memory" erstmals berechnet.

[allg. Anf.: F-LAH\_RxSWIN-589]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die CRC32-Checksumme eines Bootloader-Datensatzes wird mit dem positiven Response für die Routine "0x0202-Check Memory" persistent gespeichert.

[allg. Anf.: F-LAH\_RxSWIN-590]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die CRC32-Checksumme eines Bootloader-Datensatzes wird mit jedem Wechsel auf Kl.15-Ein oder Startup erneut berechnet und persistent gespeichert.

[allg. Anf.: F-LAH\_RxSWIN-1001]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Steuergeräte müssen für die Berechnung der CRC-Checksumme von Bootloader-Datensätzen (bei Startup und mit der Routine 0x0202-Check Memory) den Algorithmus CRC32 gemäß Dokument /6/ verwenden.

Hinweis: Damit erfolgt die Berechnung im Steuergerät identisch zur offboardseitigen Berechnung in z.B. den IT-Systemen.

[allg. Anf.: F-LAH\_RxSWIN-662]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der Einzel-Hashwert für die Bootloader-Datensätze wird mit dem Empfang des Requests für die Routine "0x0253-Calculate\_integrity\_validation\_data" mit der ControlOption [Type\_of\_calculation] = 0x00 berechnet.

[allg. Anf.: F-LAH\_RxSWIN-1006]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der Einzel-Hashwert für alle Bootloader-Datensätze wird aus den konkatenierten Tupeln aller Datensätze, bestehend aus der Datensatznummer, dem Datensatzname, der Datensatzversion und der CRC32-Checksumme entsprechend Dokument /7/, berechnet.

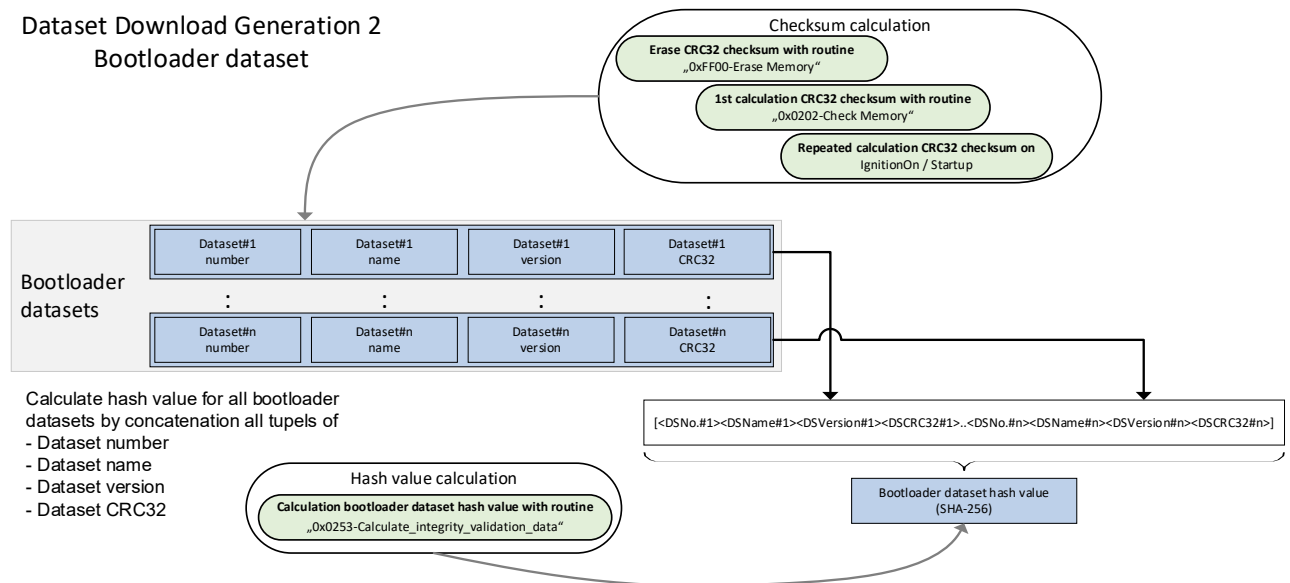
[Prozess-Anf.: F-LAH\_RxSWIN-1007]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Datensatznummer wird aus dem XML-Tag <SOURCE-START-ADDRESS>...</SOURCE-START-ADDRESS> entnommen. Einstellige Datensatznummern sind hierbei im höhenwertigen Byte um 0x00 auf eine Länge von 2 Byte zu erweitern (0x01 -> 0x00 01).

[I: F-LAH\_RxSWIN-591]

**Abbildung 4-7 CRC32-Checksummenberechnung über Bootloader Datensätze der Generation 2**



[I: F-LAH\_RxSWIN-998]

Dieser Ablauf gilt auch für die Berechnung und das Auslesen von Einzel-Hashwerten mittels der Routine "0x0254- Calculate\_individual\_hash\_value".

#### 4.1.2.1.3 Gesamt-Hashwertberechnung der Konfigurationsdaten

[allg. Anf.: F-LAH\_RxSWIN-347]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

In DK2/DK2F/DK2FV/DK3/DK3V/DK4/DK4V/SWCL-Systemen wird anhand der Liste aus dem DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" der Hashwert der Konfigurationsdaten berechnet.

[allg. Anf.: F-LAH\_RxSWIN-622]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Routine "0x0253-Calculate\_integrity\_validation\_data", mit der ControlOption [Type\_of\_calculation] = 0x00, stößt die Berechnung des Hashwertes der Konfigurationsdaten, über alle in der Liste aufgeführten DataIdentifier und Datensatznummern, in der durch die Liste vorgegeben Reihenfolge, an.

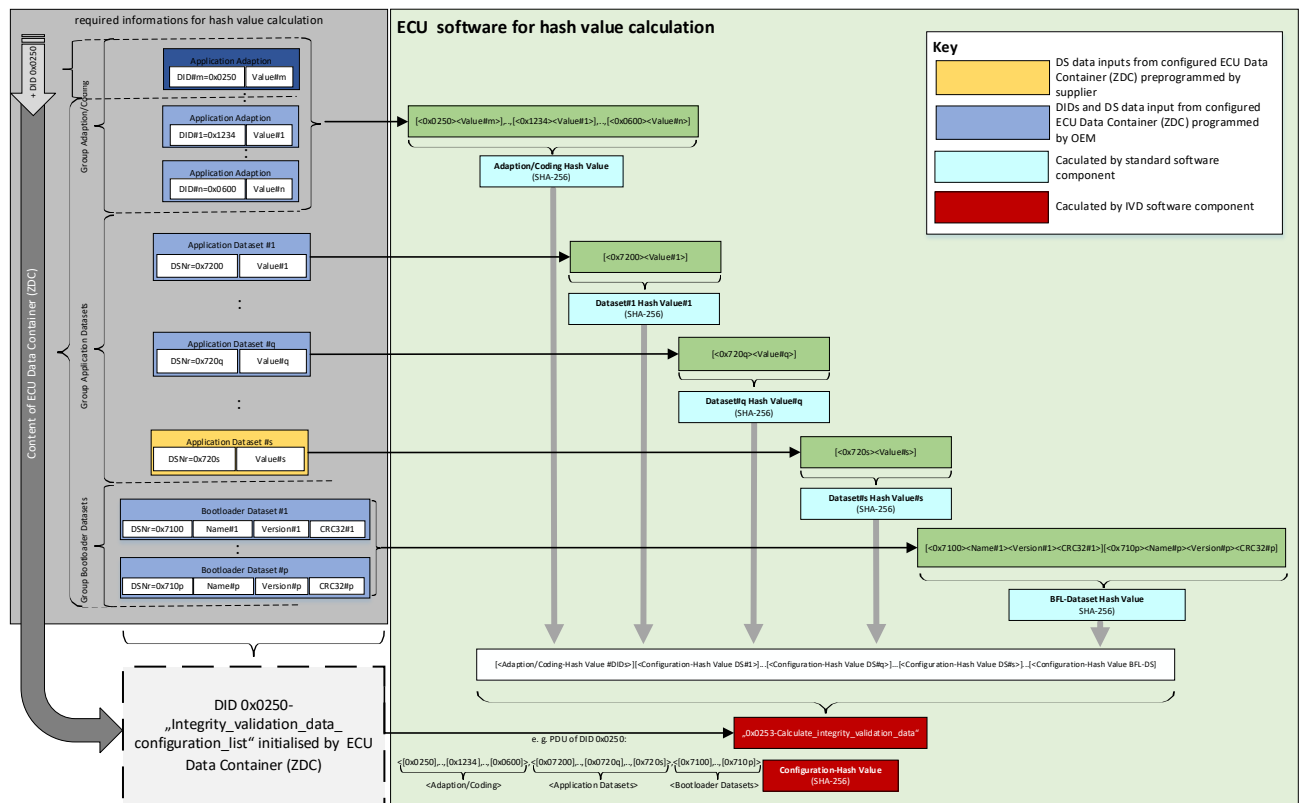
[allg. Anf.: F-LAH\_RxSWIN-344]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der auszugebende Gesamt-Hashwert aller Konfigurationsdaten ergibt sich als Hashwert aller konkatinierten Einzel-Hashwerte.

[I: F-LAH\_RxSWIN-236]

**Abbildung 4-8 Berechnung des Gesamt-Hashwertes der Konfigurationsdaten (bei DSDL Gen. 2)**



[I: F-LAH\_RxSWIN-392]

DSNr = Datensatznummer (2 Byte Hex) gemäß Dokument /7/ 0x7100 - 0x71FF für Bootloader Datensätze und 0x7200 - 0x72FF für Applikations-Datensätze

DID = DataIdentifizier gemäß Dokument /1/

Value = Dateninhalt der Anpassungen bzw. der Codierungen oder bei Datensätzen die Programmierdaten zwischen den Tags <DATEN>...</DATEN>

Hash = Hashwert des Applikations-Datensatzes (SHA-256 mit 32 Byte Länge)

Version = Bootloader-Datensatz Version entsprechend Identifikation "0xF1B1-VW\_Application\_data\_set\_identification" mit der entsprechenden Datensatznummer gemäß Dokument /2/

Name = Bootloader-Datensatz Name entsprechend Identifikation "0xF1B3-VW\_Data\_set\_name" mit der entsprechenden Datensatznummer gemäß Dokument /2/

[I: F-LAH\_RxSWIN-667]

DS-Adr = Datensatzadresse (2 Byte Hex) gemäß Dokument /11/

DID = DataIdentifizier gemäß Dokument /1/

Value = Dateninhalt der Anpassungen bzw. der Codierungen oder bei Datensätzen die Programmierdaten zwischen den Tags <DATEN>...</DATEN>

Hash = Hashwert des Applikations-Datensatzes (SHA-256 mit 32 Byte Länge)

#### 4.1.2.2 Anforderungen an Prozesse

[Prozess-Anf.: F-LAH\_RxSWIN-663]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Alle zur Konfigurations-Hashwert Berechnung relevanten Konfigurationsdaten müssen in den Diagnosedaten-Tabellen des BT-LAH gekennzeichnet werden und im ZDC enthalten sein.

[Prozess-Anf.: F-LAH\_RxSWIN-1131]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Im ZDC ist es nicht zulässig über die PKZZ-Steuerung Diagnoseobjekte gezielt so zu steuern, dass sie nur in der Produktion, in der Produktion und im Kundendienst oder nur im Kundendienst geschrieben werden. Alle Diagnoseobjekte müssen immer in der Produktion und im Kundendienst geschrieben werden.

[Prozess-Anf.: F-LAH\_RxSWIN-479]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Reihenfolge der Werte der Dataldentifier und Datensatznummern im Dataldentifier "0x0250-Integrity\_validation\_data\_configuration\_list" wird durch den ZDC-Bedatungsprozess ermittelt und in das Steuergerät eingebracht.

[Prozess-Anf.: F-LAH\_RxSWIN-1011]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Im ZDC werden die Konfigurationsdaten in folgender Weise gruppiert (geclustert)

- Anpassungen / Codierung
- Applikations-Datensätze
- Bootloader-Datensätze

[Prozess-Anf.: F-LAH\_RxSWIN-1012]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Reihenfolge der Gruppierungen ist systemspezifisch durch den jeweilige BTV im ZDC festzulegen.

[Prozess-Anf.: F-LAH\_RxSWIN-1014]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Reihenfolge der Dataldentifier bzw. der Datensatznummern innerhalb einer Gruppierungen ist systemspezifisch durch den jeweiligen BTV im ZDC festzulegen.

[Prozess-Anf.: F-LAH\_RxSWIN-1013]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die, durch den ZDC vorgegebene, Reihenfolge der Gruppierungen und die vorgegebene Reihenfolge der Dataldentifier und Datensatznummern innerhalb der Gruppierungen bestimmen die Reihenfolge der Dataldentifier und Datensatznummern im Dataldentifier "0x0250-Integrity\_validation\_data\_configuration\_list" und damit den Umfang und die Reihenfolge der Berechnung des Konfigurations-Hashwertes.

Hinweis: Die reale Konfigurationsreihenfolge (Steuergeräte Schreibvorgang Reihenfolge) darf von der vorgegebenen Reihenfolge der Dataldentifier und Datensatznummern aus dem ZDC abweichen und muss über den Inbetriebnahmeablauf festgelegt werden.

[Prozess-Anf.: F-LAH\_RxSWIN-1015]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Im Dataldentifier "0x0250-Integrity\_validation\_data\_configuration\_list" ist der erste Eintrag der Gruppe der Dataldentifier für Anpassungen / Codierungen der Dataldentifier 0x0250 selbst.

[Prozess-Anf.: F-LAH\_RxSWIN-1125]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Wenn im ZDC die Gruppierung für Anpassungen / Codierung nicht enthalten ist, muss der Dataldentifier "0x0250-Integrity\_validation\_data\_configuration\_list" als erster Dataldentifier im Dataldentifier selbst aufgeführt sein.

[Prozess-Anf.: F-LAH\_RxSWIN-1126]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Im DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" dürfen DataIdentifier bzw. Datensatznummern nicht mehrfach enthalten sein.

[Prozess-Anf.: F-LAH\_RxSWIN-962]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Über den DID "0x0250-Integrity\_validation\_data\_configuration\_list" wird die Reihenfolge der DataIdentifier und Datensätze für die Berechnung des Konfigurations-Hash vorgegeben.

[Prozess-Anf.: F-LAH\_RxSWIN-771]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Berechnung des Soll-Konfigurations-Hashwert erfolgt im IT-System analog der Berechnung im Diagnose-Server.

[Prozess-Anf.: F-LAH\_RxSWIN-955]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der verwendete Algorithmus zur Berechnung der Hashwerte ist in dem IT-System (hier: version42) an der Steuergeräteversion zu dokumentieren.

#### 4.1.2.2.1 Anforderungen an ZDC

[Prozess-Anf.: F-LAH\_RxSWIN-1166]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Im ZDC werden die Konfigurationsdaten wie folgt gruppiert:

[Prozess-Anf.: F-LAH\_RxSWIN-1167]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Anpassungen / Codierung:  
alle Modus-K Teile im ZDC ohne <RDIDENTIFIER>: 0x7200 bis 0x72FF

[Prozess-Anf.: F-LAH\_RxSWIN-1168]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Applikations-Datensätze (DSDL2):  
alle Modus-K Teile im ZDC mit <RDIDENTIFIER>: 0x7200 bis 0x72FF

[Prozess-Anf.: F-LAH\_RxSWIN-1169]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Bootloader-Datensätze (DSDL2):  
alle Modus-F Teile im ZDC mit <START-ADR>: 0x7100 bis 0x71FF

[Prozess-Anf.: F-LAH\_RxSWIN-1150]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Ein ZDC muss die folgenden Kriterien erfüllen:

[Prozess-Anf.: F-LAH\_RxSWIN-1151]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Jeder Modusteil muss eine der oben aufgeführten 3 erlaubten Gruppierungen zugeordnet werden können.

[Prozess-Anf.: F-LAH\_RxSWIN-1152]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Die Reihenfolge der Gruppierung im ZDC ist beliebig (systemspezifisch durch den ZDC-Ersteller festzulegen).

[Prozess-Anf.: F-LAH\_RxSWIN-1153]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Die Reihenfolge der Modusteile innerhalb einer Gruppierung im ZDC ist beliebig (systemspezifisch durch den ZDC-Ersteller festzulegen).

[Prozess-Anf.: F-LAH\_RxSWIN-1154]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Die Gruppierungen müssen in sich geschlossen und konsistent sein, ein Alternieren der Modusteile oder verschiedener Gruppenelemente ist nicht zulässig.

[Prozess-Anf.: F-LAH\_RxSWIN-1155]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Eine enthaltene Gruppierung im ZDC darf nur genau einmal vorhanden sein.  
Möglicher Prüfansatz: Wird ein Gruppierungswechsel erkannt, muss geprüft werden ob die neue Gruppierung bereits einmal aufgetreten war.

[Prozess-Anf.: F-LAH\_RxSWIN-1157]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Nicht jede Gruppierung muss in einem ZDC enthalten sein.  
Hinweis: Gruppierungen sind im ZDC nur enthalten, wenn die Funktionalität im Steuergerät umgesetzt ist.

[Prozess-Anf.: F-LAH\_RxSWIN-1158]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- In einem ZDC können maximal 3 Gruppierungen enthalten sein.



- [Prozess-Anf.: F-LAH\_RxSWIN-1159]*

*[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]*

  - In einem ZDC muss mindestens eine Gruppierung enthalten sein.
- [Prozess-Anf.: F-LAH\_RxSWIN-1160]*

*[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]*

  - Modus A, Modus C und Modus P sind im ZDC für UDS-Systeme verboten.
- [Prozess-Anf.: F-LAH\_RxSWIN-1161]*

*[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]*

  - Für Modus F sind nur Bootloader-Datensätze (DSDL2) erlaubt
- [Prozess-Anf.: F-LAH\_RxSWIN-1162]*

*[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]*

  - Der Wert von <RDIDENTIFIER> ist immer 2 Byte lang.
- [Prozess-Anf.: F-LAH\_RxSWIN-1163]*

*[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]*

  - Der Wert von <START-ADR> ist immer 2 Byte lang.
- [Prozess-Anf.: F-LAH\_RxSWIN-1164]*

*[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]*

  - Diese Prüfkriterien gelten nicht für RxSWIN Datencontainer/-generatoren.
- [Prozess-Anf.: F-LAH\_RxSWIN-1165]*

*[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]*

Wenn Herstellercode erlaubt sind, sind die oben aufgeführten Prüfungen für jeden Herstellercode separat durchzuführen.

## 4.2 Anforderungen an DK4-Systeme auf Basis Q-LAH 80127 bis Version 4.0

[I: F-LAH\_RxSWIN-425]

Für DK4-Systeme bis 80127 v4.0 (ohne Sammel-Dataldentifizier) sind für Integration Validation Data von DK2-Systemen systemspezifische Dataldentifizier in einem separaten Dataldentifizier-Bereich reserviert.

[I: F-LAH\_RxSWIN-735]

Beispiel zur systemspezifischen Bestimmung des Identifikationsservice (Dataldentifizier und ODX-Bezeichnung):

DID-Wertebereich für Programmierungs-Hashwert: 0xA800 - 0xA9FF

SubSystemNode (SSN): 0x01

Longname Teil 1: Control\_unit\_for\_wiper\_motor

Trennzeichen: "\_"

Longname Teil 2: Programming\_hash

ODX-Bildungsregel: Longnamekombination Teil 1 + Trennzeichen + Longname Teil 2 = "Control\_unit\_for\_wiper\_motor\_Programming\_hash"

DID-Bildungsregel: Offset 0xA800 + SSN = "0xA801"

Hinweis: Diese Bildungsregeln gelten bis auf einem geänderten Offset von 0xAA00 auch für den Konfigurations-Hashwert.

#### 4.2.1 Programmierungs-Hashwert

[allg. Anf.: F-LAH\_RxSWIN-399]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

DK4-low-Systeme mit unterlagerten DK2F-Systemen müssen den Identifikationsservice "Slave\_x\_programming\_hash" unterstützen.

[allg. Anf.: F-LAH\_RxSWIN-484]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der Wert des konkreten Identifikationsservice "Slave\_x\_programming\_hash" ergibt sich für jedes DK2F-System mit einer SSN kleiner/gleich 0x1FF wie folgt: DataIdentifier 0xA800 + SubSystemNo-deAddress (SSN).

[allg. Anf.: F-LAH\_RxSWIN-400]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

DK4-Systeme müssen die Daten der "Slave\_x\_programming\_hash"-DIDs bei den unterlagerten DK2F-Systemen über den standardisierten DataIdentifier "0x0249-Programming\_hash" abfragen.

[allg. Anf.: F-LAH\_RxSWIN-485]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Bei DK2F-Systemen mit einer SSN kleiner/gleich 0x1FF kann die Abfrage auch über die individuellen DataIdentifier erfolgen.

[I: F-LAH\_RxSWIN-488]

Hinweis: Bei DK2F-Systemen mit einer SSN größer 0x1FF müssen zwingend Sammel-DataIdentifier im DK4-Low-System verwendet werden.

[allg. Anf.: F-LAH\_RxSWIN-529]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der Identifikationsservice "Slave\_x\_programming\_hash" ist nur lesbar und darf mittels Service WriteDataByIdentifier (2Ehex) nicht schreibbar sein.

## 4.2.2 Konfigurations-Hashwert

[allg. Anf.: F-LAH\_RxSWIN-396]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

DK4-low-Systeme mit unterlagerten DK2/DK2F-Systemen müssen den Identifikationsservice "Slave\_x\_configuration\_hash" unterstützen.

[allg. Anf.: F-LAH\_RxSWIN-486]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der Wert des konkreten Identifikationsservice "Slave\_x\_configuration\_hash" ergibt sich für jedes DK2/DK2F-System mit einer SSN kleiner/gleich 0x1FF wie folgt: DataIdentifier 0xAA00 + SubSystemNodeAddress (SSN).

[allg. Anf.: F-LAH\_RxSWIN-397]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

DK4-low-Systeme müssen die Daten der "Slave\_x\_configuration\_hash"-DIDs bei den unterlagerten DK2/DK2F-Systemen über den standardisierten DataIdentifier "0x0245-Configuration\_hash" abfragen.

[allg. Anf.: F-LAH\_RxSWIN-487]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Bei DK2/DK2F-Systemen mit einer SSN kleiner/gleich 0x1FF kann die Abfrage auch über die individuellen DataIdentifier erfolgen.

[I: F-LAH\_RxSWIN-489]

Hinweis: Bei DK2/DK2F-Systemen mit einer SSN größer 0x1FF müssen zwingend Sammel-DataIdentifier im DK4-Low-System verwendet werden.

[allg. Anf.: F-LAH\_RxSWIN-528]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der Identifikationsservice "Slave\_x\_configuration\_hash" ist nur lesbar und darf mittels Service WriteDataByIdentifier (2Ehex) nicht schreibbar sein.

## 4.3 Anforderungen an DK2/DK2F/DK2FV-Systeme

### 4.3.1 Programmierungs-Hashwert

[I: F-LAH\_RxSWIN-186]

Über den Dataldentifizier "0x0249-Programming\_hash" wird die Integrity Validation Data der Programmierung eines DK2F/DK2FV-Servers ausgegeben.

[allg. Anf.: F-LAH\_RxSWIN-190]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der Dataldentifizier "0x0249-Programming\_hash" ist nur lesbar und darf mittels Service WriteDataByldentifizier (2Ehex) nicht schreibbar sein.

[allg. Anf.: F-LAH\_RxSWIN-191]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Das Starten der Routine "0xFF00-Erase Memory", die einen logischen Block mit Instruction-Code adressiert, entsprechend Dokument /6/ erfordert das Löschen der bestehenden Integrity Validation Data im DK2F/DK2FV-System für den Dataldentifizier "0x0249-Programming\_hash".

[allg. Anf.: F-LAH\_RxSWIN-744]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der Empfang des Requests mit dem Service ReadDataByldentifizier (22hex) zum Lesen des Dataldentifizier "0x0249-Programming\_hash" des Hashwertes eines DK2F/DK2FV Diagnose-Servers bewirkt eine Neuberechnung des Hashwertes in dem DK2F/DK2FV-System. Solange die Neuberechnung nicht abgeschlossen ist, muss der Request mit einem NRC "0x78-RequestCorrectlyReceived-ResponsePending" beantwortet werden. Erst nach dem Abschluss der Neuberechnung wird der Request positiv beantwortet.

[allg. Anf.: F-LAH\_RxSWIN-738]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

In DK2F-Systemen mit einer SSN kleiner oder gleich 0x1FF, erfordert das Starten der Routine "0xFF00-Erase Memory", die einen logischen Block mit Instruction-Code adressiert, entsprechend Dokument /6/ das Löschen der bestehenden Integrity Validation Data in dem individuelle Dataldentifizier 0xA800 + SSN "VW\_slave\_programming\_hash".

[allg. Anf.: F-LAH\_RxSWIN-626]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Mit dem Starten der Routine "0xFF00-Erase Memory" für einen logischen Block mit Instruction-Code, entsprechend Dokument /6/, muss die CRC32-Checksumme des adressierten Blocks gelöscht werden.

[allg. Anf.: F-LAH\_RxSWIN-627]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Nach der positiven Prüfung, dass der logische Block mit der Routine "0x0202-Check Memory" entsprechend Dokument /6/ fehlerfrei übertragen wurde, muss die CRC32-Checksumme des logischen Blocks neu berechnet werden.

[allg. Anf.: F-LAH\_RxSWIN-719]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Mit dem positiven Response für die Routine "0x0202-Check Memory", die einen logischen Block mit Instruction-Code adressiert, wird die berechnete CRC32-Checksumme des logischen Blocks persistent gespeichert.

[allg. Anf.: F-LAH\_RxSWIN-490]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Zur Gewährleistung der Abwärtskompatibilität der DK2F-Systeme mit einer SSN kleiner/gleich 0x1FF zu DK4-Systemen auf Basis VW80127 V4.0 muss zusätzlich zu dem Dataldentifizier "0x0249-Programming\_hash" auch der individuelle Dataldentifizier aus dem Bereich 0xA800 bis 0xA9FF mit gleichem Inhalt/Funktionalität unterstützt werden.

[/ : F-LAH\_RxSWIN-736]

Hinweis: Der generische DataIdentifier "0x0249-Programming\_hash" im DK2F/DK2FV-System erfordert den Sammel-DataIdentifier "0x0247-Slave\_list\_programming\_hash" im übergeordneten DK4-low-System.

### 4.3.2 Konfigurations-Hashwert

[I: F-LAH\_RxSWIN-491]

Über den DataIdentifier "0x0245-Configuration\_hash" wird die Integrity Validation Data der Konfigurationsdaten eines DK2/DK2F/DK2FV-Servers ausgegeben.

[allg. Anf.: F-LAH\_RxSWIN-130]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der DataIdentifier "0x0245-Configuration\_hash" ist nur lesbar und darf mittels Service WriteDataByIdentifier (2Ehex) nicht schreibbar sein.

[allg. Anf.: F-LAH\_RxSWIN-138]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der Empfang des Requests mit dem Service ReadDataByIdentifier (22hex) zum Lesen des DataIdentifier "0x0245-Configuration\_hash" des Hashwertes eines DK2/DK2F/DK2FV Diagnose-Servers bewirkt eine Neuberechnung des Hashwertes in dem DK2/DK2F/DK2FV-System. Solange die Neuberechnung nicht abgeschlossen ist, muss der Request mit einem NRC "0x78-RequestCorrectlyReceived-ResponsePending" beantwortet werden. Erst nach dem Abschluss der Neuberechnung wird der Request positiv beantwortet.

[allg. Anf.: F-LAH\_RxSWIN-739]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

In DK2F-Systemen mit einer SSN kleiner oder gleich 0x1FF, erfordert der empfangene Request mit dem Service ReadDataByIdentifier (22hex) zum Lesen des individuellen DataIdentifier 0xAA00 + SSN "VW\_slave\_configuration\_hash" die Neuberechnung des Integrity Validation Data.

[I: F-LAH\_RxSWIN-140]

Der Hashwert wird über alle in der Liste des DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" enthaltenen schreibbaren DataIdentifier berechnet. Diese Berechnung ist detailliert im Abschnitt "Integrity Validation Data / Allgemeine Anforderungen / Konfigurations-Hashwert" beschrieben.

[allg. Anf.: F-LAH\_RxSWIN-492]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Zur Gewährleistung der Abwärtskompatibilität der DK2F-Systeme mit einer SSN kleiner/gleich 0x1FF zu DK4-Systemen auf Basis VW80127 V4.0 muss zusätzlich zu dem DataIdentifier "0x0245-Configuration\_hash" auch der individuelle DataIdentifier aus dem Bereich 0xAA00 bis 0xABFF mit gleichem Inhalt/Funktionalität unterstützt werden.

[I: F-LAH\_RxSWIN-737]

Hinweis: Der generische DataIdentifier "0x0245-Configuration\_hash" im DK2F/DK2FV-System erfordert den Sammel-DataIdentifier "0x0248-Slave\_list\_configuration\_hash" im übergeordneten DK4-low-System.

#### 4.4 Standardsoftware-Modul für Integrity Validation Data

[I: F-LAH\_RxSWIN-845]

Für die Implementierung von Integrity Validation Data werden von der VOLKSWAGEN AG verschiedene SSW-Module und Adapter (u. a. für AUTOSAR v3.2, v4.3, BareMetal) und eine C-Referenzimplementierung zur Verfügung gestellt

[I: F-LAH\_RxSWIN-1140]

Das IVD-Standardsoftwaremodul ist über die Standardsoftwareabteilung der VOLKSWAGEN AG als Single-Source-Software zu beziehen

[I: F-LAH\_RxSWIN-1142]

Hinweis:

Single-Source: Es ist keine alternative Quelle, als die angegebene, für diese Standard-Software zulässig.

Optional-Source: Es könnte auch andere Quelle, als die angegebene, verwendet werden.

[allg. Anf.: F-LAH\_RxSWIN-1141]

[Bsm-E.D: nein, Bsm-E.L: nein, Bsm-E.T: nein, Bsm-O: nein, Bsm-Sa.FuSi: nein]

Der Auftragnehmer muss grundsätzlich für IVD-fähige Diagnose-Server ein IVD-SSW-Modul gemäß "Single-Source"-Modell (siehe Dokument /22/) einsetzen. Abweichungen hiervon sind mit der Diagnosefachabteilung abzustimmen.



## 4.5 Ablauf

[I: F-LAH\_RxSWIN-931]

Bei den folgenden Abbildungen handelt es sich um beispielhafte Teil-Abläufe, die nicht implementiert werden dürfen.

### 4.5.1 Beispielhaftes Auslesen aller relevanten Identifikationsdaten und Integrity Validation Data eines Diagnose-Servers

[I: F-LAH\_RxSWIN-246]

Ein Diagnose-Client liest die Identifikationen eines DK4-Systems mit unterlagerten DK2F/DK2FV-Systemen aus.

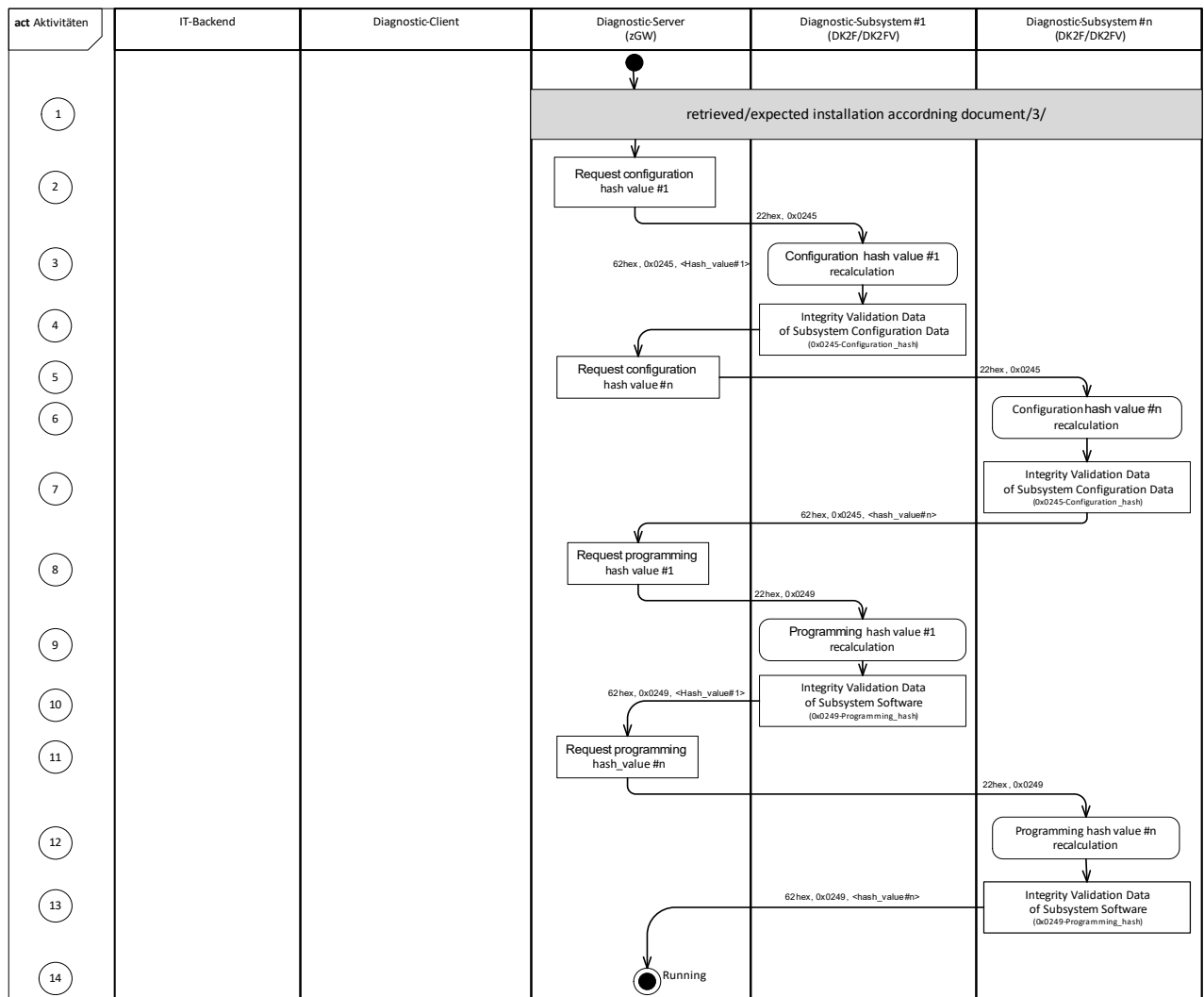
Ausgelesen werden die:

[I: F-LAH\_RxSWIN-836]

- Fahrzeug und Steuergeräte spezifischen Identifikationsdaten  
[I: F-LAH\_RxSWIN-835]
- Integrity Validation Data der Konfigurationsdaten der Anpassung/Codierung/Applikation- und Bootloader-Datensätze eines DK4-Systems  
[I: F-LAH\_RxSWIN-834]
- Integrity Validation Data der Programmierung (Instruction-Code) eines DK4-Systems  
[I: F-LAH\_RxSWIN-833]
- Integrity Validation Data der Konfigurationsdaten der Anpassung (Data) eines unterlagerten DK2F/DK2FV-Systems  
[I: F-LAH\_RxSWIN-832]
- Integrity Validation Data der Programmierung (Instruction-Code) eines unterlagerten DK2F/DK2FV-Systems

[I: F-LAH\_RxSWIN-670]

**Abbildung 4-10 Initialisierung der Sammel-Datadentifizier für Integrity Validation Data am Beispiel Gateway - Teil 1/4**



[I: F-LAH\_RxSWIN-865]

1 – Nach dem Aufstarten des zGW aktualisiert das zGW die Ist-Verbauliste mit allen erkannten unterlagerten Diagnostic-Subsystemen nach Dokument /3/, Kapitel "Erhebung der Identifikationsdaten von DK1/2/2F-Systemen".

[I: F-LAH\_RxSWIN-671]

2 – Nach der Aktualisierung der Verbauliste des zGWs fragt das zGW am unterlagerten Diagnostic-Subsystem#1 den Datadentifizier "0x0245-Configuration\_hash" für den Hashwert der Konfigurationsdaten ab.

[I: F-LAH\_RxSWIN-672]

3 – In dem Diagnostic-Subsystem#1 startet der Empfang des Requests ReadDataByIdentifizier mit dem Datadentifizier "0x0245-Configuration\_hash" die Neuberechnung des Hashwertes der Konfigurationsdaten.

[I: F-LAH\_RxSWIN-673]

4 – Das Diagnostic-Subsystem#1 sendet die Response mit dem Neuberechneten Hashwert der Konfigurationsdaten. Das zGW aktualisiert den Sammel-Datadentifizier "0x0248-Slave\_list\_configuration\_hash" anhand des empfangenen Datadentifiziers "0x0245-Configuration\_hash".

[I: F-LAH\_RxSWIN-674]

5 – Das zGW fragt am unterlagerten Diagnostic-Subsystem#n den Datadentifizier "0x0245-Configuration\_hash" für den Hashwert der Konfigurationsdaten des unterlagerten Diagnostic-Subsystems#n ab.

Hinweis: Das Abfragen der Diagnostic-Subsysteme #1 bis #n kann an den Bussen parallel erfolgen zur Zeitersparnis, hier wird nur die sequentielle Abfrage dargestellt.

[I: F-LAH\_RxSWIN-675]

6 – In dem Diagnostic-Subsystem#n startet der Empfang des Requests ReadDataByIdentifizier mit dem Datadentifizier "0x0245-Configuration\_hash" die Neuberechnung des Hashwertes der Konfigurationsdaten.

[I: F-LAH\_RxSWIN-676]

7 – Das Diagnostic-Subsystem#n sendet den Neuberechneten Hashwert der Konfigurationsdaten. Das zGW aktualisiert den Sammel-Datadentifizier "0x0248-Slave\_list\_configuration\_hash" anhand des empfangenen Datadentifiziers "0x0245-Configuration\_hash".

[I: F-LAH\_RxSWIN-677]

8 – Das zGW fragt am unterlagerten Diagnostic-Subsystem#1 den Datadentifizier "0x0249-Programming\_hash" für den Hashwert der Programmierung ab.

[I: F-LAH\_RxSWIN-710]

9 – In dem Diagnostic-Subsystem#1 startet der Empfang des Requests ReadDataByIdentifizier mit dem Datadentifizier "0x0249-Programming\_hash" die Neuberechnung des Hashwertes der Programmierungsdaten.

[I: F-LAH\_RxSWIN-679]

10 – Das unterlagerten Diagnostic-Subsystem#1 sendet den gespeicherten Hashwert der Programmierung. Das zGW aktualisiert den Sammel-Datadentifizier "0x0247-Slave\_list\_configuration\_hash" anhand des empfangenen Datadentifiziers "0x0249-Programming\_hash".

[I: F-LAH\_RxSWIN-680]

11 – Das zGW fragt am Diagnostic-Subsystem#n den Datadentifizier "0x0249-Programming\_hash" für den Hashwert der Programmierung ab.

[I: F-LAH\_RxSWIN-711]

12 – In dem Diagnostic-Subsystem#n startet der Empfang des Requests ReadDataByIdentifizier mit dem Datadentifizier "0x0249-Programming\_hash" die Neuberechnung des Hashwertes der Programmierungsdaten.

[I: F-LAH\_RxSWIN-682]

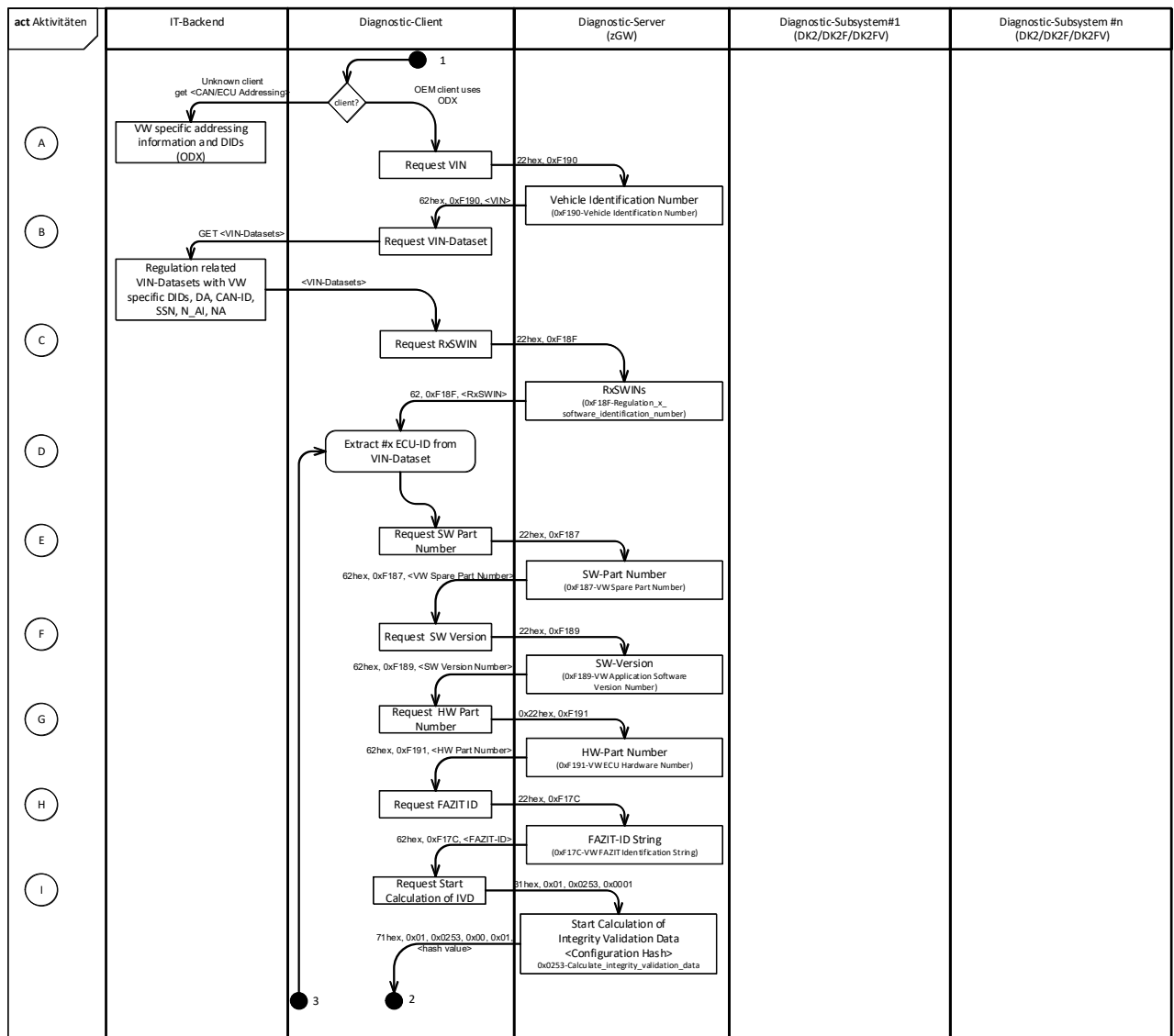
13 – Das Diagnostic-Subsystem#n sendet den gespeicherten Hashwert der Programmierung. Das zGW aktualisiert den Sammel-Datadentifizier "0x0247-Slave\_list\_programming\_hash" anhand des empfangenen Datadentifiziers "0x0249-Programming\_hash".

[I: F-LAH\_RxSWIN-683]

14 – Das zGW aktualisiert die weiteren Sammel-Datadentifizier der Diagnostic-Subsysteme.

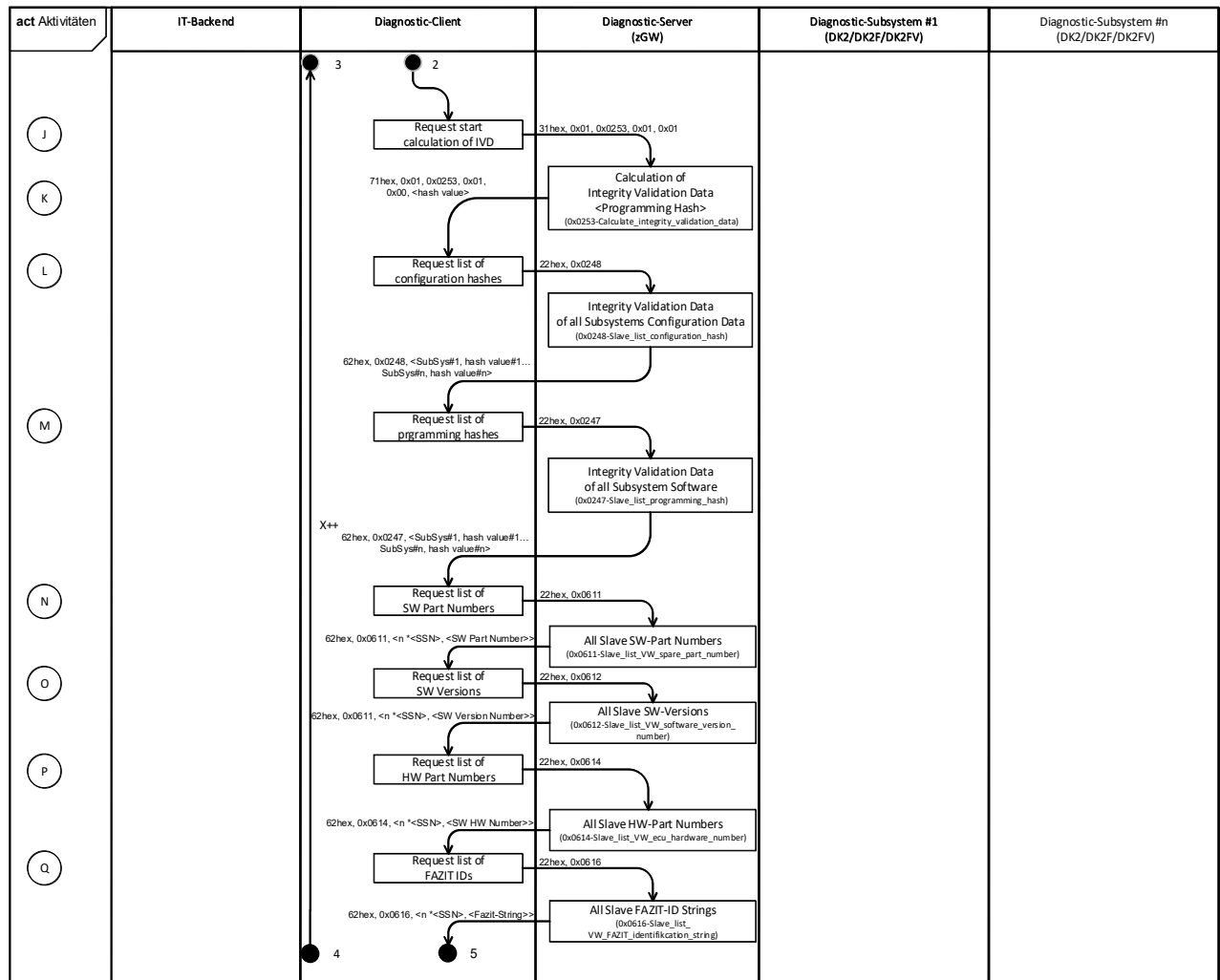
[I: F-LAH\_RxSWIN-245]

**Abbildung 4-11 Auslesen aller Identifikationsdaten und Integrity Validation Data - Teil 2/4**



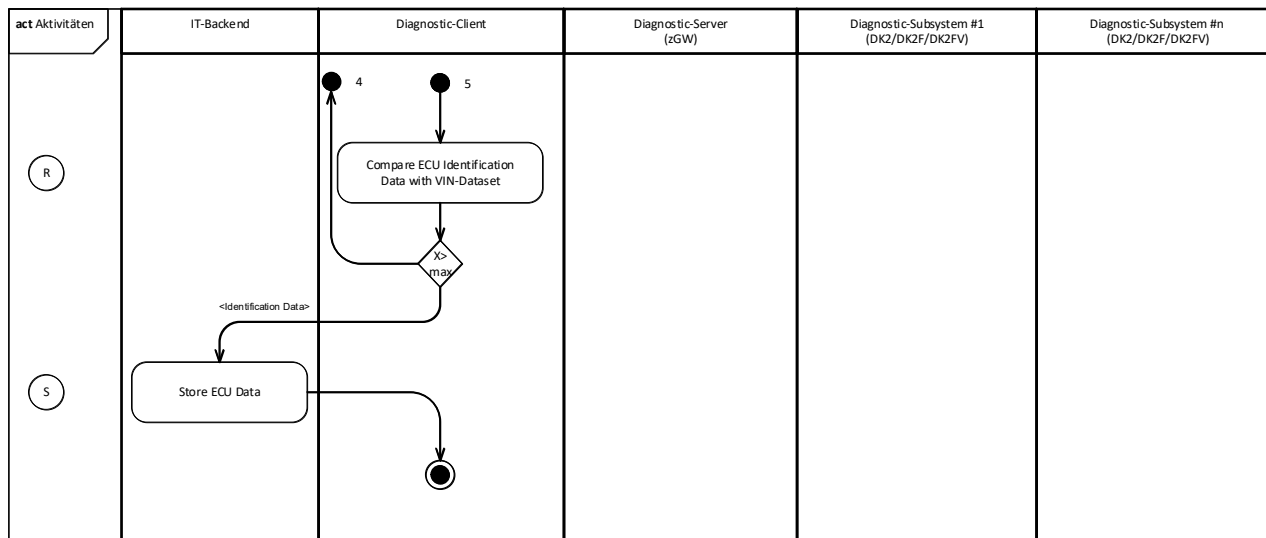
[/ F-LAH\_RxSWIN-248]

**Abbildung 4-12 Auslesen aller Identifikationsdaten und Integrity Validation Data - Teil 3/4**



[I: F-LAH\_RxSWIN-250]

**Abbildung 4-13 Auslesen aller Identifikationsdaten und Integrity Validation Data Teil 4/4**



[I: F-LAH\_RxSWIN-249]

A – Der Diagnostic-Client fragt die notwendigen Adressierungsinformationen zur Kommunikation mit dem Fahrzeug und die notwendigen DataIdentifier (DID) für die Identifikationsdaten des Fahrzeugs vom IT-Backend ab. Adressierungsinformationen sind die Diagnose-Adressen (DA), Knoten-Adressen (NA), Subsystem-Knotenadresse (SSN), Netzwerkadress-Information (N\_AI), CAN-Identifizier für Request und Response. Der Diagnostic-Client fragt die Fahrgestellnummer des aktuellen Fahrzeugs über den DataIdentifier "0xF190-Vehicle Identification Number" ab. Bei Verwendung von ODIS stehen alle notwendigen Informationen in den ODX-Daten zur Verfügung.

[I: F-LAH\_RxSWIN-251]

B – Der Diagnostic-Client fragt den fahrzeugspezifischen und regulierungsbezogenen Datensatz für die Fahrgestellnummer des Fahrzeugs am IT-Backend ab, um weitere Adressierungsinformationen, wie z. B. Diagnose-Adressen, Knoten-Adressen und DataIdentifier zu erhalten.

[I: F-LAH\_RxSWIN-252]

C – Der Diagnostic-Client fragt den DataIdentifier "0xF18F-Regulation\_x\_software\_identification\_numbers" mit der Liste aller Regulierungsnummern und SW-Identifikationsnummer am zentralen Gateway ab.

[I: F-LAH\_RxSWIN-253]

D – Der Diagnostic-Client adressiert anhand der Daten des regulierungsbezogenen Datensatz nacheinander die einzelnen im Fahrzeug verbauten Diagnostic-Server.

[I: F-LAH\_RxSWIN-254]

E – Der Diagnostic-Client liest die SW-Teilnummer über den DataIdentifier "0xF187-VW Spare Part Number" des adressierten Diagnostic-Server aus.

[I: F-LAH\_RxSWIN-255]

F – Der Diagnostic-Client liest die SW-Version über den DataIdentifier "0xF189-VW Software Version Number" des adressierten Diagnostic-Server aus.

[I: F-LAH\_RxSWIN-256]

G – Der Diagnostic-Client liest die HW-Teilnummer über den DataIdentifier "0xF191-VW ECU Hardware Number" des adressierten Diagnostic-Server aus.

[I: F-LAH\_RxSWIN-257]

H – Der Diagnostic-Client liest die FAZIT-ID über den DataIdentifier "0xF17C-VW FAZIT Identification String" des adressierten Diagnostic-Server aus.

[I: F-LAH\_RxSWIN-259]

I – Der Diagnostic-Client fragt den Hashwert der Konfigurationsdaten des adressierten Diagnostic-Server ab. Der DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" enthält dafür die Liste aller DataIdentifier und Datensatznummern über die der Hashwert der Konfigurationsdaten berechnet werden muss. Der Routin-Identifier "0x0253-Calculate\_integrity\_validation\_data" liefert in der positiven Response den Hashwert der Konfigurationsdaten des Diagnostic-Server.

[I: F-LAH\_RxSWIN-260]

J – Der Diagnostic-Client startet für den adressierten Diagnostic-Server die Berechnung des Hashwertes über die gesamte Programmierung.

[I: F-LAH\_RxSWIN-262]

K – Der Diagnostic-Client erhält für den Diagnostic-Server den neu berechneten Hashwert der Programmierung.

[I: F-LAH\_RxSWIN-263]

L – Der Diagnostic-Client fragt für den Diagnostic-Server den Sammel-DataIdentifier "0x0248-Slave\_list\_configuration\_hash", für die Hashwerte der Konfigurationsdaten aller Diagnostic-Subsysteme ab.

[I: F-LAH\_RxSWIN-264]

M – Der Diagnostic-Client fragt für den Diagnostic-Server den Sammel-DataIdentifier "0x0247-Slave\_list\_programming\_hash", für die Hashwerte der Programmierung aller Diagnostic-Subsysteme ab.

[I: F-LAH\_RxSWIN-266]

N – Der Diagnostic-Client liest die Liste aller SW-Teilnummer der Diagnostic-Subsysteme über den Sammel-DataIdentifier "0x0611-Slave\_list\_VW\_spare\_part\_number" über den adressierten Diagnostic-Server aus.

[I: F-LAH\_RxSWIN-545]

O – Der Diagnostic-Client liest die Liste aller SW-Version der Diagnostic-Subsysteme über den Sammel-DataIdentifier "0x0612-Slave\_list\_VW\_software\_version\_number" über den adressierten Diagnostic-Server aus.

[I: F-LAH\_RxSWIN-546]

P – Der Diagnostic-Client liest die Liste aller HW-Teilnummer der Diagnostic-Subsysteme über den Sammel-DataIdentifier "0x0614-Slave\_list\_VW\_ecu\_hardware\_number" über den adressierten Diagnostic-Server aus.

[I: F-LAH\_RxSWIN-267]

Q – Der Diagnostic-Client liest die Liste aller FAZIT-Identifikationen der Diagnostic-Subsysteme über den Sammel-DataIdentifier "0x0616-Slave\_list\_VW\_FAZIT\_identification\_string" über den adressierten Diagnostic-Server aus.

[I: F-LAH\_RxSWIN-270]

R – Diagnostic-Client vergleicht die Identifikationsdaten für die RxSWIN mit dem Datensatz aus dem IT-Backend und liest danach die Identifikationsdaten für das nächste Diagnostic-Server aus.

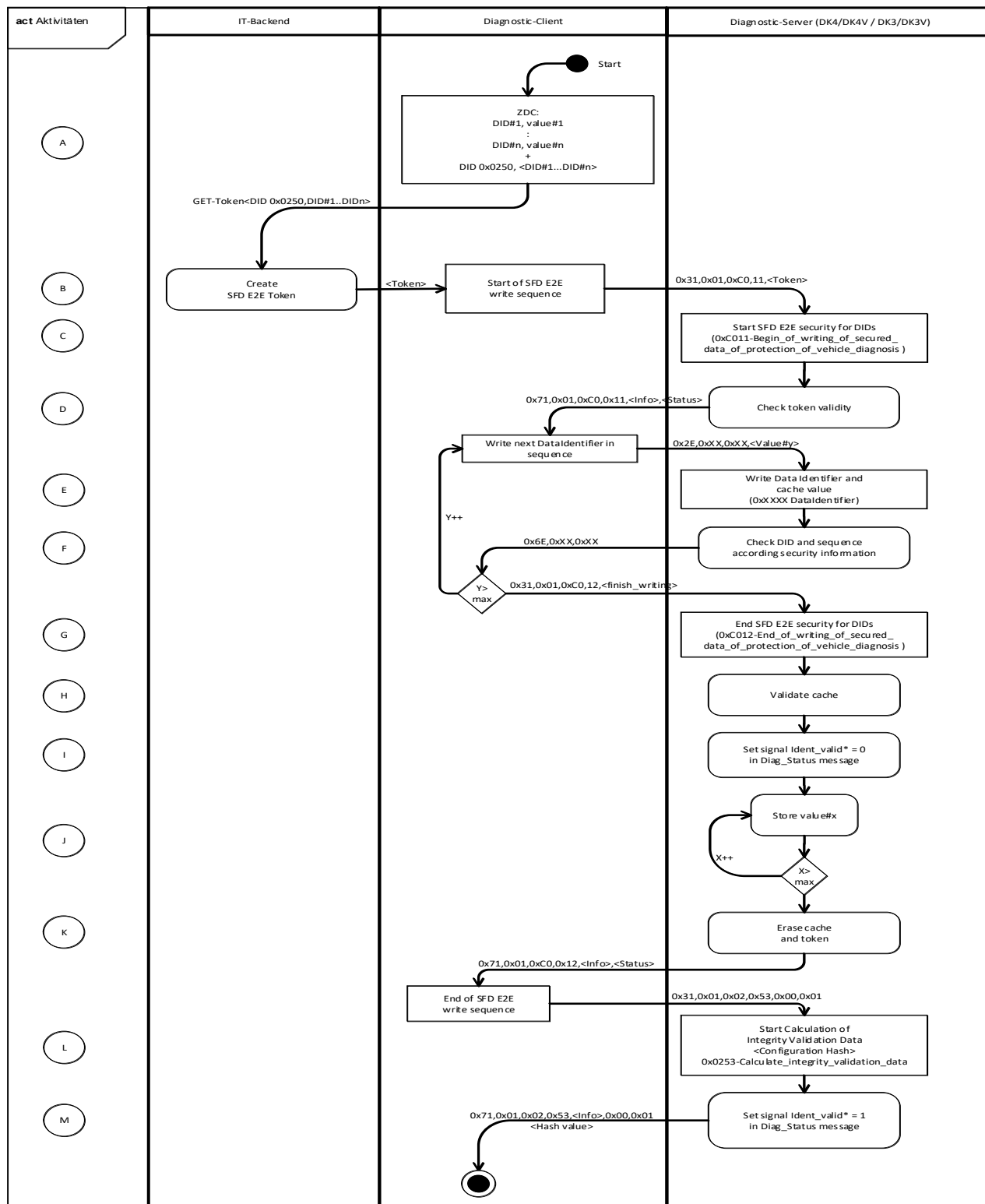
[I: F-LAH\_RxSWIN-547]

S – Der Diagnostic-Client speichert die ausgelesenen Identifikationsdaten im IT-Backend.

## 4.5.2 Beispielhaftes Schreiben von Daten in ein SFD-E2E abgesichertes DK4/DK4V bzw. DK3/DK3V System

[I: F-LAH\_RxSWIN-356]

Abbildung 4-14 Schreiben von Daten eines DK3/DK3V bzw. DK4/DK4V-Systems, Teil 1/1



\*Note: The signal „Ident\_valid“ (QLAH 80114 from version 5.6) has only informational character here and no requirement character.



[I: F-LAH\_RxSWIN-537]

A - Die für die Steuergeräte-Konfiguration notwendigen DataIdentifier werden dem Diagnostic-Client zur Verfügung gestellt. Die DataIdentifier ergeben sich durch den Bauauftrag, der den ZDC konfiguriert, hieraus resultiert auch der Inhalt des DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list". Der Diagnostic-Client fragt ein Absicherungs-Token am SFD-IT-Backend für die im konfigurierten ZDC enthaltene DataIdentifier DID#1 ... DID#n und dem berechneten DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" an.

[I: F-LAH\_RxSWIN-359]

B\* - Das SFD-IT-Backend erstellt ein Absicherungs-Token über die DataIdentifier DID#1 ... DID#n und DID 0x0250.

[I: F-LAH\_RxSWIN-360]

C\* - Der Absicherungs-Token wird mit dem Start der SFD-Routine "0xC011-Begin\_of\_writing\_of\_secured\_data\_of\_protection\_of\_vehicle\_diagnosis" an den Diagnostic-Server übertragen und leitet die Schreib-Sequenz für SFD-Ende-zu-Ende abgesicherte Daten ein.

[I: F-LAH\_RxSWIN-361]

D\* - Der Diagnostic-Server prüft die Validität des empfangenen SFD-Absicherungs-Tokens. Entsprechend der SFD-E2E Absicherung gemäß Dokument /4/ wird bei einem Empfang eines ungültigen Tokens die Absicherungsinformationen gelöscht.

[I: F-LAH\_RxSWIN-362]

E - Die DataIdentifier aus dem konfigurierten ZDC und der DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" werden mittels Service WriteDataByIdentifier (2Ehex) an den Diagnostic-Server übertragen und zwischengespeichert. Ohne SFD werden die Daten direkt in den Speicher geschrieben.

[I: F-LAH\_RxSWIN-363]

F\* - Entsprechend der SFD-E2E Absicherung aus Dokument /4/ wird die Empfangsreihenfolge der einzelnen DataIdentifier entsprechend der Absicherungsinformation überprüft. Bei positiver Prüfung antwortet der Diagnostic-Server mit einem positiven Response. Bei einem negativen Prüfergebnis werden die empfangenen DataIdentifier und die zugehörigen DataRecords sowie die Absicherungsinformationen gelöscht. Bei einem negativen Prüfergebnis antwortet der Diagnostic-Server mit einem NRC "0x22-ConditionsNotCorrect".

[I: F-LAH\_RxSWIN-364]

G\* - Die SFD-Schreibsequenz wird mit dem Start der SFD-Routine "0xC012-End\_of\_writing\_of\_secured\_data\_of\_protection\_of\_vehicle\_diagnosis" mit RoutineControlOption [0x01-Finish\_writing\_of\_secured\_data] abgeschlossen.

[I: F-LAH\_RxSWIN-365]

H\* - Die Authentizität der übertragenen DataIdentifier wird anhand der Absicherungsinformation geprüft.

[I: F-LAH\_RxSWIN-368]

I - In der Diag\_Status-Botschaft des Diagnostic-Server wird im Signal [Ident\_valid] der Wert '0' angezeigt, dass die Identifikationsdaten aktuell nicht vollständig gültig sind, d. h. die Hashwerte der Konfigurationsdaten sind gelöscht und sind noch nicht erneut berechnet worden.

[I: F-LAH\_RxSWIN-366]

J\* - Bei positiver Prüfung werden alle zwischengespeicherten DataIdentifier aus dem Zwischenspeicher sequentiell in den Zielspeicher des Diagnostic-Server geschrieben. Bei einem negativen Prüfergebnis werden alle empfangenen DataIdentifier und die zugehörigen DataRecords sowie die Absicherungsinformationen gelöscht.

[I: F-LAH\_RxSWIN-369]

K\* - Der Zwischenspeicher und die Absicherungsinformationen im dem Diagnostic-Server werden gelöscht, nachdem alle DataIdentifier in den Zielspeicher übertragen wurden.

[I: F-LAH\_RxSWIN-370]

L - Der Diagnostic-Client fragt den neuen Hashwert der Konfigurationsdaten ab. Der Aufruf der Routine "0x0253-Calculate\_integrity\_validation\_data" startet die Neuberechnung des Hashwertes der Konfigurationsdaten auf dem Diagnostic-Server. Die Hashwertberechnung erfolgt entsprechend der im DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" (siehe Kapitel "0x0250-Integrity\_validation\_data\_configuration\_list") übergebenen Liste der DataIdentifier und Datensatznummern.

[I: F-LAH\_RxSWIN-371]

M - In der Diag\_Status-Botschaft des Diagnostic-Server wird im Signal [Ident\_valid] der Wert '1' angezeigt, dass die Identifikationsdaten vollständig gültig sind, d. h. der Hashwert wurde aktualisiert.

[I: F-LAH\_RxSWIN-527]

\*Nur relevant für SFD-E2E abgesicherte Diagnose-Systeme. Nicht relevant für unverriegelte Diagnose-Systeme mit Gruppenfreischaltung in der Produktion, vor ZP8.

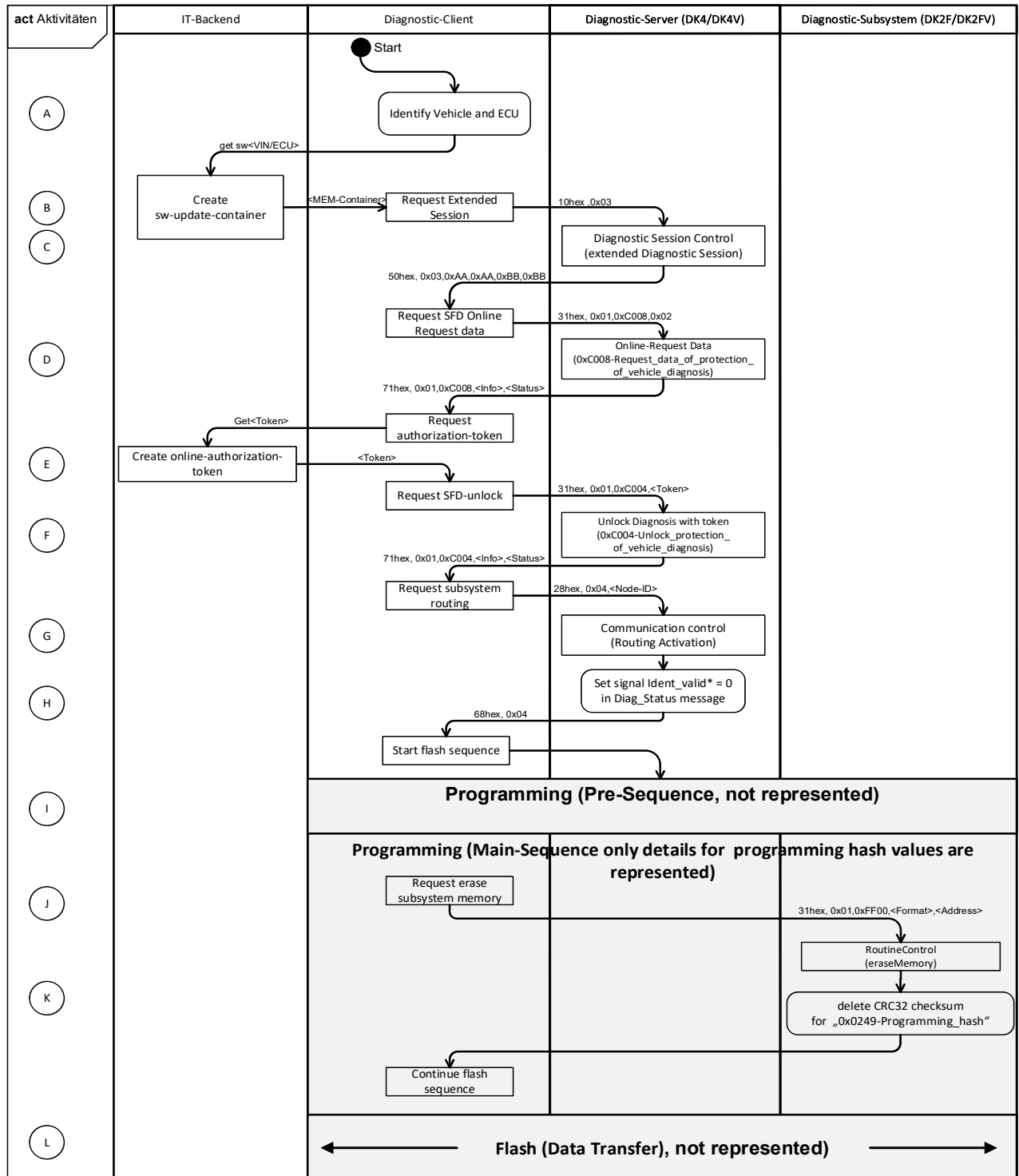
### 4.5.3 Beispielhafte Programmierung eines DK2F/DK2FV-Systems

[I: F-LAH\_RxSWIN-303]

Ablauf eines Software-Updates eines DK2F/DK2FV System über ein DK4/DK4V-System durch einen Diagnose-Client.

[I: F-LAH\_RxSWIN-305]

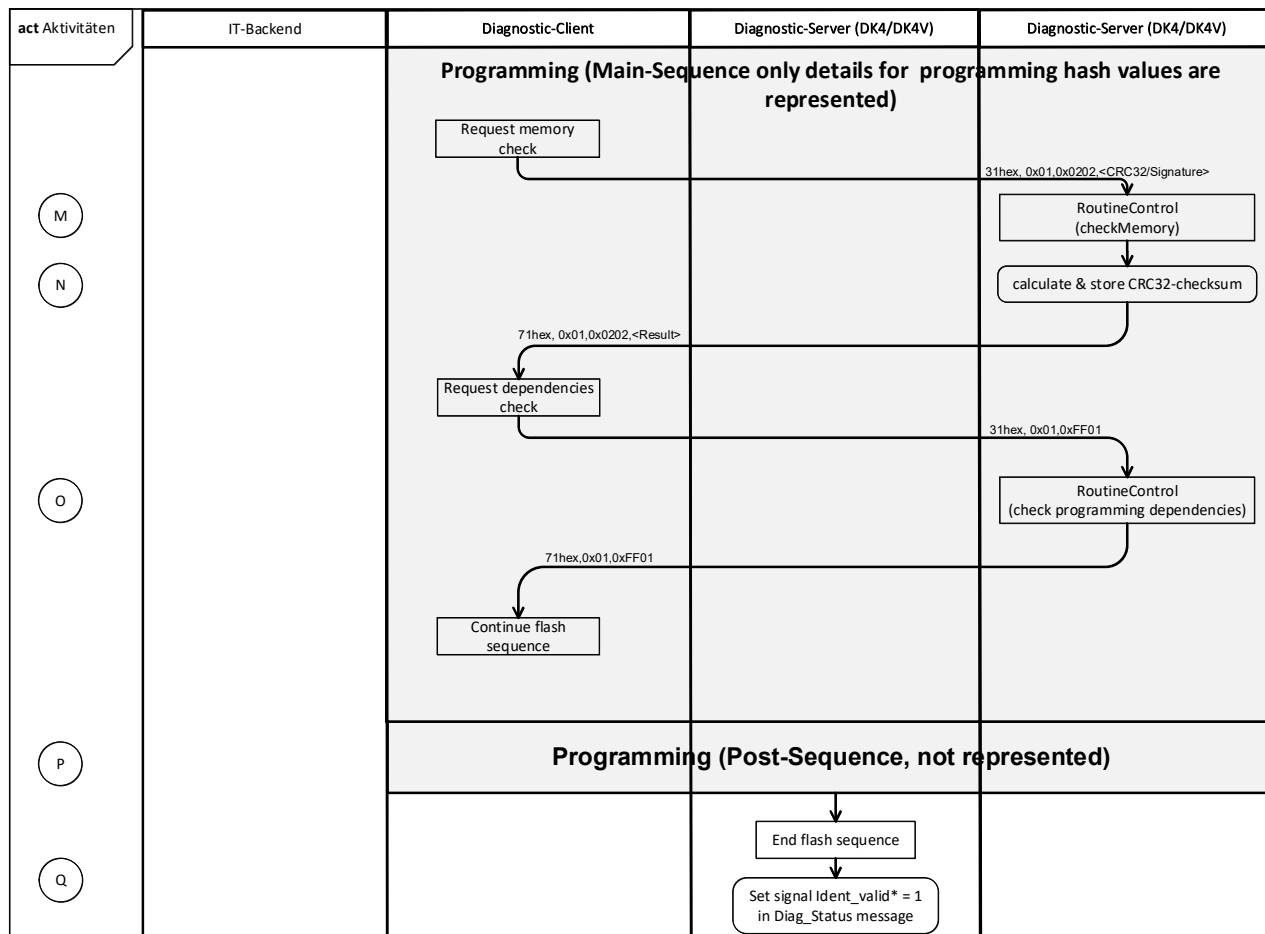
Abbildung 4-15 Beispielhafte Programmierung DK2F/DK2FV-System, Teil 1/2



\*Note: The signal „Ident\_valid“ (document QLAH 80114 from version 5.6) has only informational character here and no requirement character.

[I: F-LAH\_RxSWIN-306]

**Abbildung 4-16 Beispielhafte Programmierung eines DK2F/DK2FV-System, Teil 2/2**



\*Note: The signal „Ident\_valid“ (QLAH 80114 from version 5.6) has only informational character here and no requirement character.

[I: F-LAH\_RxSWIN-307]

A - Der Diagnostic-Client identifiziert das aktuelle Fahrzeug anhand der Fahrgestellnummer und das zu aktualisierende Steuergerät anhand seiner Identifikationsdaten.

[I: F-LAH\_RxSWIN-310]

B - Der Diagnostic-Client lädt das entsprechende Software-Update-Container, welches durch das IT-Backend zusammengestellt wurde, herunter.

[I: F-LAH\_RxSWIN-311]

C - Der Diagnostic-Client fordert für den Diagnostic-Server den Sessionwechsel in die Extended-Diagnostic-Session 0x03 an.

[I: F-LAH\_RxSWIN-312]

D - Der Diagnostic-Client fordert von dem Diagnostic-Server die Anfragedaten für eine SFD Online Freischaltung an.

[I: F-LAH\_RxSWIN-313]

E - Der Diagnostic-Client fragt am SFD-Backend mit den Anfragedaten ein Freischalt-Token an.

[I: F-LAH\_RxSWIN-314]

F - Der Diagnostic-Client schaltet den Diagnostic-Server mit dem Freischalt-Token frei, um das Routing im Diagnostic-Server über die Diagnose aktivieren zu können.

[I: F-LAH\_RxSWIN-315]

G - Der Diagnostic-Client aktiviert in dem Diagnostic-Server das Routing der Diagnose-Botschaften zu dem Diagnostic-Subsystem.

[I: F-LAH\_RxSWIN-523]

H - Das DK4/DK4V-System signalisiert über die Diag\_Status-Botschaft im Signal [Ident\_valid] = 0, dass die Steuergeräte Identifikation nicht mehr gültig ist.

[I: F-LAH\_RxSWIN-316]

I - Programmierablauf Pre-Sequenz gemäß Dokument /6/.

[I: F-LAH\_RxSWIN-318]

J - Der Diagnostic-Client fordert das Löschen eines logischen Blocks des Diagnostic-Subsystems an.

[I: F-LAH\_RxSWIN-319]

K - Das Löschen des Speichers über die Routine „0xFF00-Erase Memory“ bewirkt, dass gleichzeitig die CRC32-Checksumme für den zu löschenden Block gelöscht wird.

[I: F-LAH\_RxSWIN-320]

L - Der Diagnostic-Client führt die Flash-Sequenz mit dem Diagnostic-Subsystem durch.

[I: F-LAH\_RxSWIN-321]

M - Der Diagnostic-Client fordert das Diagnostic-Subsystem nach der Flash-Sequenz auf, den korrekten Empfang des Speicherblocks zu überprüfen.

[I: F-LAH\_RxSWIN-322]

N - Der Diagnostic-Server routet den Request zur Überprüfung des Speichers an das Diagnostic-Subsystem direkt weiter. Das Diagnostic-Subsystem berechnet die CRC32-Checksumme des logischen Blockes und speichert sie persistent ab.

[I: F-LAH\_RxSWIN-323]

O - Der Diagnostic-Client fordert das Diagnostic-Subsystem nach der Flash-Sequenz auf, den korrekt empfangenen Speicherblock hinsichtlich Kompatibilität/Plausibilität zu überprüfen.

[I: F-LAH\_RxSWIN-324]

P - Programmierablauf Post-Sequenz gemäß Dokument /6/.

[I: F-LAH\_RxSWIN-525]

Q - Der Diagnostic-Server signalisiert über die Diag\_Status-Botschaft dass die Steuergeräte Identifikation gültig ist.

## 5 Diagnoseobjekte

[I: F-LAH\_RxSWIN-54]

Die folgende Tabelle zeigt die Übersicht der SUMS-relevanten Diagnosefunktionen bezogen auf die Diagnoseklasse.

[allg. Anf.: F-LAH\_RxSWIN-180]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-1 Diagnosefunktion und Diagnoseklassen**

|  | DK4-high | DK4-low | DK4V-low | DK4 <sup>1)</sup> | DK3   | DK3V  | DK2   | DK2F  | DK2FV | SWCL  | Bemerkung   |
|--|----------|---------|----------|-------------------|-------|-------|-------|-------|-------|-------|---|
| Diagnoseobjekt/-funktion   |          |         |          |                   |       |       |       |       |       |       |   |
| RxSWIN im DID "0xF18F-Regulation_x_software_identification_numbers"  | x        | Nolmp   | Nolmp    | Nolmp             | Nolmp | Nolmp | Nolmp | Nolmp | Nolmp | Nolmp | Nur einmal pro Fahrzeug im zentralen Gateway umzusetzen   |
| SW-Teilnummern im DID "0xF187-VW Spare Part Number"  | x        | x       | x        | x                 | x     | x     | x     | x     | x     | x     |   |
| SW-Version im DID "0xF189-VW Software Version Number"  | x        | x       | x        | x                 | x     | x     | x     | x     | x     | x     |   |
| HW-Teilnummer im DID "0xF191-VW ECU Hardware Number"   | x        | x       | Nolmp    | x                 | x     | Nolmp | x     | x     | Nolmp | Nolmp |   |
| HW-Version im DID "0xF1A3-VW ECU Hardware Version Number"  | x        | x       | Nolmp    | x                 | x     | Nolmp | x     | x     | Nolmp | Nolmp |   |
| Fahrgestellnummer im DID "0xF190-Vehicle Identification Number"  | (x)      | (x)     | (x)      | (x)               | (x)   | (x)   | Nolmp | Nolmp | Nolmp | Nolmp | VIN nur bei WFS-Teilnehmer, DoIP-Server und zGW   |
| Konfigurations-Hashwert im DID "0245-Configuration_hash"   | Nolmp    | Nolmp   | Nolmp    | Nolmp             | Nolmp | Nolmp | x     | x     | x     | Nolmp | Hashwert der Konfiguration. Diagnose-Server, die nur flashbar und nicht konfigurierbar sind, können diesen DID mit NRC 0x31 ablehnen.   |
| Sammel-DID "0x0248-Slave_list_configuration_hash" für "0x0245-Configuration_hash"                          | Nolmp    | x       | x        | Nolmp             | Nolmp | Nolmp | Nolmp | Nolmp | Nolmp | Nolmp | Bei DK4-high nur mit unterlagerten DK2-Systemen   |
| Programmierungs-Hashwert im DID "0249-Programming_hash"  | Nolmp    | Nolmp   | Nolmp    | Nolmp             | Nolmp | Nolmp | Nolmp | x     | x     | Nolmp | Hashwert des Programms  |
| Sammel-DID 0x0247-Slave_list_programming_hash für "0x0249-Programming_hash"                                | Nolmp    | x       | x        | Nolmp             | Nolmp | Nolmp | Nolmp | Nolmp | Nolmp | Nolmp | Bei DK4-high nur mit unterlagerten DK2F-Systemen  |
| 0x0251_Write_generic_to_sub_system   | Nolmp    | x       | x        | x                 | Nolmp | Nolmp | Nolmp | Nolmp | Nolmp | Nolmp | Bei DK4-high nur mit unterlagerten DK2-Systemen   |
| Liste zur Hashwertberechnung von Konfigurationsdaten im DID "0x0250-List of configuration data identifier" | x        | x       | x        | x                 | x     | x     | x     | x     | x     | x     | Bei DK2/DK2F: Bedienung erfolgt über den Busmaster/Host mit 0x0251  |
| Berechnung der Integrity Validation Data mit RID "0x0253-Calculate_integrity_validation_data"              | x        | x       | x        | x                 | x     | x     | Nolmp | Nolmp | Nolmp | x     | Hashwert der Programm und Daten   |
| Berechnung der Einzel-Hashwerte mit RID "0x0254-Calculate_individual_hash_value"                           | x        | x       | x        | x                 | x     | x     | Nolmp | Nolmp | Nolmp | x     | Einzel-Hashwert der Konfigurationsdaten   |
| Neuberechnung der CRC32-Checksumme mit RID "0x0544-Verify_partial_software_checksum"                       | x        | x       | x        | x                 | x     | x     | Nolmp | x     | x     | x     | CRC32 Berechnung der logischen Blöcke für embedded Systemen die keine systemspezifische Berechnung des Hashwertes der Programmierung umsetzen. (SOLL-Hashwert nicht im Updatecontainer enthalten)                           |
| Vollständige Neuberechnung der IVD mit RID "0x029A-Calculate_module_hash_value"                            | x        | x       | x        | x                 | x     | x     | Nolmp | x     | x     | x     | Hashwert der Programmierung bei nonEmbedded/filebasierten Systemen und bei embedded Systemen die eine systemspezifische Berechnung des Hashwertes der Programmierung umsetzen (SOLL-Hashwert im Updatecontainer enthalten). |
| Programmierungs-Hashwert für Q-LAH 80127 <V5.x-Subsysteme (DID-Bereich 0xA800 - 0xA9FF)                    | Nolmp    | (x)     | (x)      | x                 | Nolmp | Nolmp | Nolmp | x     | Nolmp | Nolmp | Nur für DK4 mit DK2F in Bestandsarchitekturen (individueller DID für 0608er-Identifikation nach 80127 <V5.x)  |
| Konfigurations-Hashwert für Q-LAH 80127 <V5.x-Subsysteme (DID-Bereich 0xAA00 - 0xABFF)                     | Nolmp    | (x)     | (x)      | x                 | Nolmp | Nolmp | x     | x     | Nolmp | Nolmp |   |

[I: F-LAH\_RxSWIN-855]

<sup>\*)</sup> = DK4 entsprechend Q-LAH 80127 bis V4.0 (ohne Unterteilung in DK4-low/DK4-high)

[I: F-LAH\_RxSWIN-856]

x = beinhaltet

[I: F-LAH\_RxSWIN-857]

(x) = optional

## 5.1 DataIdentifizier

[allg. Anf.: F-LAH\_RxSWIN-933]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-2 Übersicht der Eigenschaften der DataIdentifizier**

| Diagnoseobjekte (DID)                               | SID   | Security Level                                  | DiagnosticSession |                       |                             |
|---|-------|---|-------------------|-----------------------|-----------------------------|
|   |       |   | Default Session   | NonDefaultSession     |                             |
|   |       |   |                   | ECUProgrammingSession | Extended-Diagnostic-Session |
|   |       |   | 0x01              | 0x02                  | 0x03                        |
| 0x0249-Programming_hash                             | 22hex | NoImp   | Available         | notAvailable          | Available                   |
|   | 2Ehex | NoImp   | notAvailable      | notAvailable          | notAvailable                |
| 0x0247-Slave_list_programming_hash                  | 22hex | NoImp   | Available         | notAvailable          | Available                   |
|   | 2Ehex | NoImp   | notAvailable      | notAvailable          | notAvailable                |
| 0x0245-Configuration_hash                           | 22hex | NoImp   | Available         | notAvailable          | Available                   |
|   | 2Ehex | NoImp   | notAvailable      | notAvailable          | notAvailable                |
| 0x0248-Slave_list_configuration_hash                | 22hex | NoImp   | Available         | notAvailable          | Available                   |
|   | 2Ehex | NoImp   | notAvailable      | notAvailable          | notAvailable                |
| 0xF18F-Regulation_x_software_identification_number  | 22hex | NoImp   | Available         | notAvailable          | Available                   |
|   | 2Ehex | SFD-E2E   | notAvailable      | notAvailable          | Available                   |
| 0x0250-Integrity_validation_data_configuration_list | 22hex | NoImp   | Available         | notAvailable          | Available                   |
|   | 2Ehex | SFD-E2E *)                                      | notAvailable      | notAvailable          | Available                   |
| 0x0251-Write_generic_to_sub_system                  | 22hex | NoImp   | notAvailable      | notAvailable          | notAvailable                |
|   | 2Ehex | Old platform:<br>C1<br>New platform:<br>SFD-E2E | notAvailable      | notAvailable          | Available                   |

[I: F-LAH\_RxSWIN-1064]

\*) Wenn das Steuergerät keine SFD-Ende-zu-Ende-Absicherung unterstützt, ist kein Zugriffsschutz notwendig.

[I: F-LAH\_RxSWIN-936]

C1 - Das Zugriffsschutzverfahren (sofern vorhanden) des DK4-low ist zu verwenden.

### 5.1.1 0xF1A3-VW ECU Hardware Version Number

[I: F-LAH\_RxSWIN-664]

Dieser DataIdentifier kennzeichnet die Hardwareversion eines Steuergerätes.

[allg. Anf.: F-LAH\_RxSWIN-701]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Folgende Änderungen gegenüber Q-LAH 80125 bis V5.9 sind zu beachten:

[allg. Anf.: F-LAH\_RxSWIN-806]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Entfall von:

- Die Hardwareversion ist nicht steuerungsrelevant für den Fahrzeugproduktionsprozess oder den Ersatzteilhandel.
- Ob eine software- und hardwarekompatible Änderung durch die Hardwareversion dokumentiert wird, liegt im Ermessen des BSB in Absprache mit dem Auftragnehmer.

[allg. Anf.: F-LAH\_RxSWIN-805]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Neu:

Folgende Änderung sind durch die Hardwareversion zu dokumentieren:

- Änderung von aktiven Bauteilen (z. B.  $\mu$ C,  $\mu$ P, RAM, Flash, ASIC)
- Änderungen an der Platine z. B. bzgl. EMV, Stromaufnahme
- Änderungen am funktionalen Verhalten: Z. B. optionaler Vorhalt wird bestückt
- Änderung beeinflusst Schnittstellen

#### 5.1.1.1 Anforderungen an Systeme in neuen Plattformen

[allg. Anf.: F-LAH\_RxSWIN-231]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Entgegen der VW80125, ist die Hardwareversion aus dem DataIdentifier "0xF1A3-VW ECU Hardware Version Number" nur noch „user optional“.



## 5.1.2 0x0249-Programming\_hash

[allg. Anf.: F-LAH\_RxSWIN-449]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-3 Aufbau DataRecord DataIdentifier 0x0249-Programming\_hash**

|                |   |
|----------------|---|
| <b>DID</b>     | <b>0x0249</b>   |
| Bezeichnung    | Programming_hash  |
| Beschreibung   | Dieser DataIdentifier beinhaltet die Integrity Validation Data der Programmdateien. Die Berechnung des Hashwertes erfolgt über die gesamte Software. Er muss bei Fahrzeugstillstand und im Zustand "keine Fahrbereitschaft" auslesbar sein. In anderen Fahrzeugzuständen (z.B. aktive Regeleingriffe) können Steuergerätespezifische Randbedingungen gelten. Die Signale zur Fahrbereitschaft und Geschwindigkeit müssen architekturabhängig gemäß gültiger Datenfestlegung verwendet werden. |
| Convention     | alle Server   |
| Diagnoseklasse | DK2F/DK2FV  |
| Session        | APP: 0x01 (R), 0x03 (R), 0x40 (R)<br>BLF: NoImp   |
| SecurityLevel  | NoImp   |
| Changing       | APP   |
| Format         | <[Programming_hash] 32-Byte-Hex>  |
| Range          | [Programming_hash] 00..00hex - FF..FFhex  |
| Init           | entfällt, immer verfügbar   |
| Beispiel       | -   |
| Datenkategorie | Analysedaten  |
| eBZD           | M   |

### 5.1.3 0x0247-Slave\_list\_programming\_hash

[allg. Anf.: F-LAH\_RxSWIN-443]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-4 Aufbau DataRecord DataIdentifier 0x0247-Slave\_list\_programming\_hash**

|                |   |
|----------------|---|
| <b>DID</b>     | <b>0x0247</b>   |
| Bezeichnung    | Slave_list_programming_hash   |
| Beschreibung   | Dieser DataIdentifier beinhaltet alle Integrity Validation Data über Instructioncode der unterlagerten DK2F/DK2FV-Systeme.<br>Es gelten folgende Festlegungen: <ul style="list-style-type: none"> <li>• Der jeweilige Inhalt entspricht dem Parameter [Programming_hash] des DataIdentifier "0x0249-Programming_hash"</li> <li>• Dem jeweiligen Inhalt muss die jeweilige SubSystemNodeAddress vorangestellt werden.</li> </ul> |
| Convention     | Busmaster   |
| Diagnoseklasse | DK4-low /DK4V-low   |
| Session        | APP: 0x01 (R), 0x03 (R), 0x40 (R)<br>BLF: NoImp   |
| SecurityLevel  | NoImp   |
| Changing       | APP   |
| Format         | <[NumberOfExpectedSubSystemIdentification] 1-Byte-Hex>+<br><[NumberOfRetrievedSubSystemIdentification] 1-Byte-Hex>+<br><n-mal<br><[SubSystemNodeAddress] 2-Byte-Hex>+<br><[Programming_hash] 32-Byte-Hex><br>><br>(n: Anzahl der DK2F/DK2FV-Systeme)  |
| Range          | [NumberOfExpectedSubSystemIdentification]: 00hex - FFhex<br>[NumberOfRetrievedSubSystemIdentification]: 00hex - FFhex<br>[SubSystemNodeAddress]: 00 00hex - FF FFhex<br>[Programming_hash]: siehe DID 0x0249  |
| Init           | entfällt, immer verfügbar   |
| Beispiel       | -   |
| Datenkategorie | Analysedaten  |
| eBZD           | NoImp   |

## 5.1.4 0x0245-Configuration\_hash

[allg. Anf.: F-LAH\_RxSWIN-445]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-5 Aufbau DataRecord DataIdentifier 0x0245-Configuration\_hash**

|                |   |
|----------------|---|
| <b>DID</b>     | <b>0x0245</b>   |
| Bezeichnung    | Configuration_hash  |
| Beschreibung   | Dieser DataIdentifier beinhaltet die Integrity Validation Data der Konfigurationsdaten. Er muss bei Fahrzeugstillstand und im Zustand "keine Fahrbereitschaft" auslesbar sein. In anderen Fahrzeugzuständen (z.B. aktive Regeleingriffe) können Steuergeräte-spezifische Randbedingungen gelten. Die Signale zur Fahrbereitschaft und Geschwindigkeit müssen architekturabhängig gemäß gültiger Datenfestlegung verwendet werden. |
| Convention     | alle Server   |
| Diagnoseklasse | DK2/DK2F/DK2FV  |
| Session        | APP: 0x01 (R), 0x03 (R), 0x40 (R)<br>BLF: NoImp   |
| SecurityLevel  | NoImp   |
| Changing       | APP   |
| Format         | <[Configuration_hash] 32-Byte-Hex>  |
| Range          | [Configuration_hash] 00..00hex - FF..FFhex  |
| Init           | entfällt, immer verfügbar   |
| Beispiel       | -   |
| Datenkategorie | Analysedaten  |
| eBZD           | M   |

### 5.1.5 0x0248-Slave\_list\_configuration\_hash

[allg. Anf.: F-LAH\_RxSWIN-447]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-6 Aufbau DataRecord DataIdentifier 0x0248-Slave\_list\_configuration\_hash**

|                |  |
|----------------|--|
| <b>DID</b>     | <b>0x0248</b>  |
| Bezeichnung    | Slave_list_configuration_hash  |
| Beschreibung   | Dieser DataIdentifier beinhaltet alle Integrity Validation Data über Konfigurationsdaten der unterlagerten DK2/DK2F/DK2FV-Systeme.<br>Es gelten folgende Festlegungen: <ul style="list-style-type: none"> <li>• Der jeweilige Inhalt entspricht dem Parameter [Configuration_hash] des DataIdentifier "0x0245-Configuration_hash".</li> <li>• Dem jeweiligen Inhalt muss die jeweilige SubSystemNodeAddress vorangestellt werden.</li> </ul> |
| Convention     | Busmaster  |
| Diagnoseklasse | DK4-low /DK4V-low  |
| Session        | APP: 0x01 (R), 0x03 (R), 0x40 (R)<br>BLF: NoImp  |
| SecurityLevel  | NoImp  |
| Changing       | APP  |
| Format         | <[NumberOfExpectedSubSystemIdentification] 1-Byte-Hex>+<br><[NumberOfRetrievedSubSystemIdentification] 1-Byte-Hex>+<br><n-mal<br><[SubSystemNodeAddress] 2-Byte-Hex>+<br><[Configuration_hash] 32-Byte-Hex><br>><br>(n: Anzahl der DK2/DK2F/DK2FV-Systeme)   |
| Range          | [NumberOfExpectedSubSystemIdentification]: 00hex - FFhex<br>[NumberOfRetrievedSubSystemIdentification]: 00hex - FFhex<br>[SubSystemNodeAddress]: 00 00hex - FF FFhex<br>[Configuration_hash]: siehe DID 0x0245   |
| Init           | entfällt, immer verfügbar  |
| Beispiel       | -  |
| Datenkategorie | Analysedaten   |
| eBZD           | NoImp  |

### 5.1.6 0xF18F-Regulation\_x\_software\_identification\_numbers

[allg. Anf.: F-LAH\_RxSWIN-14]

[BsM-E.D.: nein, BsM-E.L.: nein, BsM-E.T.: nein, BsM-O.: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-7 Aufbau DataRecord DataIdentifier 0xF18F-Regulation\_x\_software\_identification\_numbers**

|                |   |
|----------------|---|
| <b>DID</b>     | <b>0xF18F</b>   |
| Bezeichnung    | Regulation_x_software_identification_numbers  |
| Beschreibung   | Dieser DataIdentifier beinhaltet die Liste der Software-Identifikationsnummern von z.B. UNECE- oder in chinesischen Normen definierten GB/T-Funktionen, die in einem Fahrzeug verfügbar sind. Diese Liste ist fahrzeugspezifisch. Sämtliche Nutzinhalte wie [Length_of_RxSWIN], [Regulation_identification], [Separation_character] und [Software_identification] dürfen vom Diagnose-Server nicht plausibilisiert werden und müssen unverändert ausgegeben werden. |
| Convention     | Steuergerät mit zentralem Diagnosezugang  |
| Diagnoseklasse | DK4-high  |
| Session        | APP: 0x01 (R), 0x03 (R/W), 0x40 (R/W)<br>BLF: NoImp   |
| SecurityLevel  | SFD-E2E (W)   |
| Changing       | DIAG  |
| Format         | <n-mal<br><[Length_of_RxSWIN] 1-Byte-Hex>+<br><[Regulation_identification] m-Byte-ASCII, variabel>+<br><[Separation_character] 1-Byte-ASCII>+<br><[Software_identification] 1..11-Byte-ASCII, variabel><br>><br>(n: Anzahl der RxSWINs, variabel;<br>m: Anzahl der Bytes für die Regulierungs-ID, variabel)   |
| Range          | [Length_of_RxSWIN] 00-FFhex<br>[Regulation_identification] 21-7Ahex<br>[Separation_character] 20hex (ASCII "SPACE")<br>[Software_identification] 30-39hex, 41-5Ahex, 61-7A<br>Andere Wert sind ISO-reserved   |
| Init           | 2D 2D 2D 2D 2Dhex ( ' - - - - - ' )   |
| Beispiel       | 0F 52 30 37 39 20 76 30 35 37 34 31 37 35 33 61 13 47 42 2F 54 33 36 30 34 37 20<br>76 30 34 33 36 39 38 35 32<br>Anzahl RxSWIN = 2<br>RxSWIN #1:<br>Length_of_RxSWIN: 0x0F = 15 Byte<br>Regulation_identification: R079<br>Software_identification: v05741753a<br>RxSWIN #2:<br>Length_of_RxSWIN: 0x13 = 19 Byte<br>Regulation_identification: GB/T36047<br>Software_identification: v04369852   |
| Datenkategorie | Prozessparameter  |
| eBZD           | M   |

[I: F-LAH\_RxSWIN-70]

Die Gesamtlänge der RxSWIN soll mindestens 50 Regulierungen und die dazugehörigen regulierungsbezogenen Software-Identifikationen beinhalten können.

[allg. Anf.: F-LAH\_RxSWIN-71]

[BsM-E.D.: nein, BsM-E.L.: nein, BsM-E.T.: nein, BsM-O.: nein, BsM-Sa.FuSi: nein]

Es ergibt sich daraus eine Mindestgröße des Speichers für den DataIdentifier "0xF18F-Regulation\_x\_software\_identification\_numbers" von 1000 Byte.

[allg. Anf.: F-LAH\_RxSWIN-67]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Das Lesen des Dataidentifiers "0xF18F-Regulation\_x\_software\_identification\_numbers" über den Service ReadDataByIdentifier (22hex) darf nicht über ein Zugriffsschutzverfahren geschützt werden.

[allg. Anf.: F-LAH\_RxSWIN-1076]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Das Schreiben des Dataidentifiers "0xF18F-Regulation\_x\_software\_identification\_numbers" muss über SFD-Ende-zu-Ende-Absicherung entsprechend Dokument /4/ erfolgen.

[allg. Anf.: F-LAH\_RxSWIN-1133]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der Dataidentifier "0xF18F-Regulation\_x\_software\_identification\_numbers" darf nicht durch die Routine "0x03E7-Reset\_to\_factory\_setting" zurückgesetzt werden.

#### 5.1.6.1 Anforderungen an Prozesse

[Prozess-Anf.: F-LAH\_RxSWIN-72]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Für nicht im Bauauftrag des Fahrzeuges vorhandene UNECE-Funktionen wird keine RxSWIN geschrieben. D. h. für nicht in dem konkreten Fahrzeug vorhandene UNECE-Funktionen darf es keinen Listeneintrag im Dataidentifier "0xF18F-Regulation\_x\_software\_identification\_numbers" mit einer RxSWIN geben.

[Prozess-Anf.: F-LAH\_RxSWIN-74]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der Dataidentifier "0xF18F-Regulation\_x\_software\_identification\_numbers" besitzt eine fahrzeug-spezifische Länge. Der Bedatungs-Prozess muss dafür sorgen, dass der Dataidentifier "0xF18F-Regulation\_x\_software\_identification\_numbers" entsprechend der Anforderung F-LAH\_RxSWIN-72 berechnet wird.

[I: F-LAH\_RxSWIN-555]

Hinweis:

Die Umsetzung kann z. B. durch einen JAVA- oder OTX-Job erfolgen.

#### 5.1.6.2 Anforderungen an IT-Systeme

[Prozess-Anf.: F-LAH\_RxSWIN-220]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Die Zusteuerung des Datencontainers mit der Liste der RxSWINs erfolgt in der Datenlogistik über ein, dem Gateway-Steuergerät zugeordneten, DK2V-System.

[Prozess-Anf.: F-LAH\_RxSWIN-803]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Hinweis: Hierbei handelt es sich nicht um ein DK2V-System im Sinne der Dokumente /3/ bzw. /12/. Dieses DK2V-System beinhaltet lediglich eine Diagnoseadresse, welche zur Steuerung der Daten in der Logistikkette notwendig ist.

### 5.1.7 0x0250-Integrity\_validation\_data\_configuration\_list

[allg. Anf.: F-LAH\_RxSWIN-1061]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Steuergeräte, die Integrity Validation Data für die Programmierung und/oder für die Konfiguration umgesetzt haben, müssen den DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" umsetzen.

[allg. Anf.: F-LAH\_RxSWIN-349]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-8 Aufbau DataRecord DataIdentifier 0x0250-Integrity\_validation\_data\_configuration\_list**

|                       |   |
|-----------------------|---|
| <b>DID</b>            | <b>0x0250</b>   |
| <b>Bezeichnung</b>    | Integrity_validation_data_configuration_list  |
| <b>Beschreibung</b>   | Dieser DataIdentifier beinhaltet die Liste der Identifier (DataIdentifier und Datensatznummern, die für die Hashwert-Berechnung von Konfigurationsdaten verwendet werden.   |
| <b>Convention</b>     | Alle Server   |
| <b>Diagnoseklasse</b> | DK2/DK2F/DK2FV, DK3/DK3V, DK4/DK4V, SWCL  |
| <b>Session</b>        | APP: 0x01 (R), 0x03 (R/W *)), 0x40 (R/W *)<br>BLF: NoImp  |
| <b>SecurityLevel</b>  | SFD-E2E *)  |
| <b>Changing</b>       | DIAG  |
| <b>Format</b>         | <<[Number_of_identifiers] 2-Byte-Hex>+<br>n-mal<[Identifier] 2-Byte-Hex><br>(n: Anzahl der Identifier)  |
| <b>Range</b>          | [Number_of_identifiers]<br>00 00hex = Kein DataIdentifier oder keine Datensatznummer vorhanden, z. B. wenn nur IVD für die Programmierung unterstützt wird.<br>00 01hex - FF FFhex<br>[Identifier] 00 00-FF FFhex |
| <b>Init</b>           | 00 00hex  |
| <b>Beispiel</b>       | 00 07 71 01 71 02 02 50 12 43 98 67 72 02 72 01<br>Anzahl Identifier = 7<br>DSNo#1: 0x7101<br>DSNo#2: 0x7102<br>DID#1: 0x0250<br>DID#2: 0x1243<br>DID#3: 0x9867<br>DSNo#1: 0x7202<br>DSNo#2: 0x7201               |
| <b>Datenkategorie</b> | Fahrzeugparameter / Prozessparameter **)  |
| <b>eBZD</b>           | NoImp   |

[allg. Anf.: F-LAH\_RxSWIN-837]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

\*) Wenn das Steuergerät keine SFD-Ende-zu-Ende-Absicherung unterstützt, ist kein Zugriffsschutz notwendig.

[allg. Anf.: F-LAH\_RxSWIN-1181]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

\*\*) Datenart = Prozessparameter wenn Steuergerät IVD nur für Programmierung unterstützt wird

[allg. Anf.: F-LAH\_RxSWIN-539]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Der Inhalt des DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" muss persistent im Steuergerät gespeichert werden.

[allg. Anf.: F-LAH\_RxSWIN-1062]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Wenn das Steuergerät den Konfigurations-Hashwert umsetzt, muss im DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" der Wert des DataIdentifier 0x0250 selbst auch als erster DataIdentifier in der Gruppe Anpassungen/Codierung enthalten sein.

[allg. Anf.: F-LAH\_RxSWIN-1065]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Unterstützt ein Steuergerät den Konfigurations-Hashwert, muss ein Request mit WriteDataByIdentifier (2Ehex) und dem DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" positiv beantwortet, wenn:

- alle aufgelisteten DataIdentifier und Datensatznummern als IVD-relevant gekennzeichnet sind.
- kein DataIdentifier oder Datensatznummer mit dem SFD-Zugriffsschutz abgesichert ist.
- kein DataIdentifier oder Datensatznummer mehrfach aufgelistet ist.
- der DataIdentifier 0x0250 als erster DataIdentifier in der Gruppe der Anpassungen/Codierung enthalten ist.
- die Gruppierungen Anpassung/Codierung, Bootloader-Datensätze und Applikationsdatensätze eingehalten werden.
- die Anzahl der DataIdentifier und Datensatznummern dem Wert im Parameter [Number\_of\_identifiers] entspricht.

[allg. Anf.: F-LAH\_RxSWIN-1063]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Wenn das Steuergerät nur den Programmierungs-Hashwert umsetzt, darf im DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" der Wert des DataIdentifier 0x0250 selbst nicht enthalten sein.

[allg. Anf.: F-LAH\_RxSWIN-1189]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Unterstützt ein Steuergerät den Konfigurations-Hashwert nicht, wird ein Request mit WriteDataByIdentifier (2Ehex) und dem DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" nur dann positiv beantwortet, wenn der Parameter [Number\_of\_identifiers] = 0x0000 ist und keine weiteren DataIdentifier oder Datensatznummern im Request enthalten sind.



### 5.1.8 0x0251-Write\_generic\_to\_sub\_system (Schreiben des DataIdentifiers 0x0250 über das DK4-low/DK4V-low-System zum unterlagerten DK2/DK2F/DK2FV-System)

[I: F-LAH\_RxSWIN-494]

Das Schreiben des DataIdentifier "0x0250-Integrity\_validation\_data\_configuration\_list" erfolgt über den Service WriteDataByIdentifier (2Ehex) und dem DataIdentifier "0x0251-Write\_generic\_to\_sub\_system" über das DK4-Low-System. Zur Adressierung des unterlagerten Zielsystems wird im 2Ehex-Request im DataRecord an den ersten beiden Bytes die SubSystemNodeAddress (SSN) verwendet. Die folgenden beiden Bytes beinhalten den DataIdentifier 0x0250 gefolgt von den Nutzdaten.

[allg. Anf.: F-LAH\_RxSWIN-615]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-9 Aufbau DataRecord DataIdentifier 0x0251-Write\_generic\_to\_sub\_system**

|                |   |
|----------------|---|
| <b>DID</b>     | <b>0x0251</b>   |
| Bezeichnung    | Write_generic_to_sub_system   |
| Beschreibung   | Dieser DataIdentifier wird zum Schreiben von Daten für Subsysteme über ein DK4-low-System verwendet.<br>Das Auslesen von Daten ist mittels dieses DataIdentifiers nicht möglich. (write only) |
| Convention     | Busmaster   |
| Diagnoseklasse | DK4/DK4-low   |
| Session        | APP: 0x03 (W), 0x40 (W)<br>BLF: NoImp   |
| SecurityLevel  | Steuergeräte in Bestandsarchitekturen: Falls vorhanden ist das bestehende Zugriffsschutzverfahren des DK4-low zu verwenden.<br>Steuergeräte in neuen Architekturen: SFD-E2E (W)               |
| Changing       | DIAG  |
| Format         | <<[Target_sub_system_node_address] 2-Byte-Hex>+<br><[Target_data_identifier] 2-Byte-Hex>+<br><n-mal [Data_record_target_data_identifier] 1-Byte-Hex>><br>(n: 2..256, variabel)                |
| Range          | [Target_sub_system_node_address] 00 00 - FF FFhex<br>[Target_data_identifier] 0x02 50<br>[Data_record_target_data_identifier] 00 - FFhex<br>Andere Wert sind VOLKSWAGEN AG-reserved           |
| Init           | kein Init im Server   |
| Beispiel       | 01 41 02 50 00 02 13 24 97 86<br>Target_sub_system_node_address: 0x0141 = Kartenleser TV Tuner<br>Target_data_identifier: 0x02 50<br>Data_record_target_data_identifier: 0x00 02 13 24 97 86  |
| Datenkategorie | Prozessparameter  |
| eBZD           | NoImp   |

### 5.1.8.1 Request Message Definition

[I: F-LAH\_RxSWIN-500]

Folgende Parameter sind zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-495]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-10 Request Message Definition**

| Data   | Description      |   | Cvt. | Value (Hex) |
|--------|------------------|---|------|-------------|
| #1     | Request SID      | WriteDataByIdentifier                       | M    | 2E          |
| #2     | DataIdentifier#1 | Write_generic_to_sub_system [Byte#1] MSB    | M    | 02          |
| #3     | DataIdentifier#2 | Write_generic_to_sub_system [Byte#2]        | M    | 51          |
| #4     | DataRecord#1     | Target_sub_system_node_address [Byte#1] MSB | M    | 00-FF       |
| #5     | DataRecord#2     | Target_sub_system_node_address [Byte#2]     | M    | 00-FF       |
| #6     | DataRecord#3     | Target_data_identifier [Byte#1] MSB         | M    | 00-FF       |
| #7     | DataRecord#4     | Target_data_identifier [Byte#2]             | M    | 00-FF       |
| #8     | DataRecord#5     | Data_record_target_data_identifier#1        | M    | 00-FF       |
| :      | :                | :   | :    | :           |
| #9+m-1 | DataRecord#5+m-1 | Data_record_target_data_identifier#m        | U    | 00-FF       |

### 5.1.8.2 Request Message Parameter Definition

[I: F-LAH\_RxSWIN-497]

Folgende Parameter sind zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-498]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

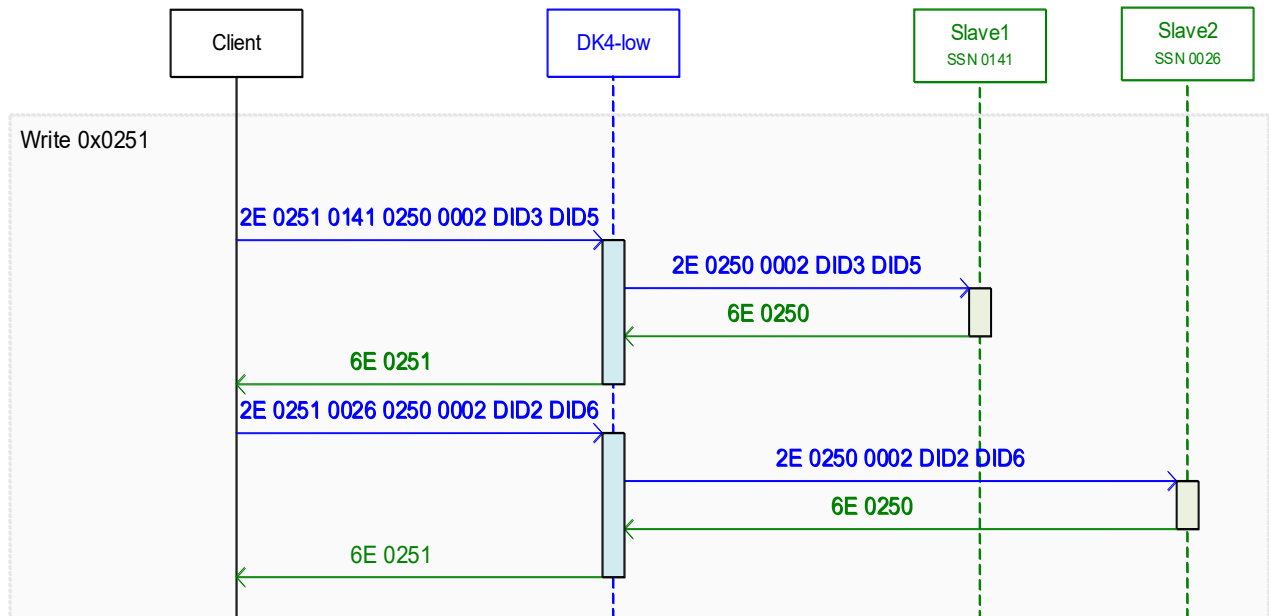
**Tabelle 5-11 Request Message Parameter Definition**

|   |
|---|
| <b>Definition</b>   |
| <b>Target_sub_system_node_address</b><br>Dieser Parameter gibt die SubSystemNodeAddress (SSN) des unterlagerten Zielsystems an.                                     |
| <b>Target_data_identifier</b><br>Dieser Parameter gibt den Wert des DataIdentifiers an. Erlaubter Wert ist hier 0x0250.<br>Andere Werte sind VOLKSWAGEN AG-reserved |
| <b>Data_record_target_data_identifier</b><br>Dieser Parameter gibt den DataRecord des Parameters [Target_data_identifier] an.                                       |

[allg. Anf.: F-LAH\_RxSWIN-475]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Abbildung 5-1 Schreiben von 0x0250-Integrity\_validation\_data\_configuration\_list**



[I: F-LAH\_RxSWIN-1182]

Hinweis: Diese Abbildung gilt nur bei nicht SFD-E2E abgesicherten Systemen.

## 5.2 Routineldentifizier

[allg. Anf.: F-LAH\_RxSWIN-934]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-12 Übersicht der Eigenschaften der Routineldentifizier**

| Diagnoseobjekte (RID)                      | SID   | Security Level | DiagnosticSession |                       |                             |
|--|-------|----------------|-------------------|-----------------------|-----------------------------|
|  |       |                | Default Session   | NonDefaultSession     |                             |
|  |       |                |                   | ECUProgrammingSession | Extended-Diagnostic-Session |
|  |       |                | 0x01              | 0x02                  | 0x03                        |
| 0x0366-Reset of all adaption               | 31hex | SFD-Basic      | NoImp             | NoImp                 | Available                   |
| 0x03E7-Reset_to_factory_setting            | 31hex | SFD-Basic      | NoImp             | NoImp                 | Available                   |
| 0x0253-Calculate_integrity_validation_data | 31hex | NoImp          | Available         | notAvailable          | Available                   |
| 0x0254-Calculate_individual_hash_value     | 31hex | NoImp          | Available         | notAvailable          | Available                   |
| 0x029A-Calculate_module_hash_value*        | 31hex | NoImp          | Available         | C1                    | Available                   |
| 0x0544-Verify_partial_software_checksum**  | 31hex | NoImp          | C2**              | Available**           | C2**                        |

[I: F-LAH\_RxSWIN-1124]

C1 - Mandatory bei DK2F/DK2FV-Systemen oder wenn die Umsetzung der Routine in der Applikation technisch nicht möglich ist.

[I: F-LAH\_RxSWIN-1135]

\*) Für nonEmbedded/filebasierte Systeme und embedded Systeme mit systemspezifischer IVD-Berechnung.

[I: F-LAH\_RxSWIN-1138]

\*\*) Nur für embedded System ohne systemspezifischer IVD-Berechnung und Update über UDS mit Flash-PDX.

[I: F-LAH\_RxSWIN-1139]

C2 - Nur wenn die Umsetzung der Routine in der Applikation technisch möglich ist.

## 5.2.1 Rücksetzen von Konfigurationsparametern

[I: F-LAH\_RxSWIN-945]

**Tabelle 5-13 Übersicht der Rücksetzbarkeit der Daten (-kategorien)**

|   | Programming-Hash | Configuration-Hash | Rücksetzen durch RID |        |
|---|------------------|--------------------|----------------------|--------|
|   |                  |                    | 0x0366               | 0x03E7 |
| <b>Programmdaten</b><br>(Program data)                    | x                | NoImp              | NoImp                | NoImp  |
| <b>Applikationsdaten</b><br>(Calibration data)            | x                | NoImp              | NoImp                | NoImp  |
| <b>Codierung</b><br>(Coding)                              | NoImp            | x                  | NoImp                | x      |
| <b>Fahrzeugparameter</b><br>(Vehicle parameters)          | NoImp            | x                  | NoImp                | x      |
| <b>Erstbedatungswerte</b><br>(Initial calibration values) | NoImp            | x                  | NoImp                | x      |
| <b>Kundenparameter</b><br>(Customer parameters)           | NoImp            | NoImp              | x                    | x      |
| <b>Werkstattparameter</b><br>(Workshop parameters)        | NoImp            | NoImp              | x                    | x      |
| <b>Prozessparameter</b><br>(Process parameters)           | NoImp            | NoImp              | NoImp                | x      |
| <b>Lernwerte</b><br>(Learned values)                      | NoImp            | NoImp              | NoImp                | x      |
| <b>Analysedaten</b><br>(Analysis data)                    | NoImp            | NoImp              | NoImp                | NoImp  |

[I: F-LAH\_RxSWIN-972]

x = beinhaltet

### 5.2.1.1 0x0366-Reset of all adaption

[I: F-LAH\_RxSWIN-970]

Dieser Routineldentifizier wird entsprechend Dokument /1/ und /8/ verwendet, um Kunden- und Werkstattparameter auf Werkseinstellungen zurückzusetzen.

[allg. Anf.: F-LAH\_RxSWIN-971]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Folgende Änderung gegenüber Q-LAH 80125 V5.3 bis V5.5 ist zu beachten:

- SFD-Zugriffsschutz (Rolle Basic) für Routineldentifizier 0x0366.

### 5.2.1.2 0x03E7-Reset\_to\_factory\_setting

[I: F-LAH\_RxSWIN-927]

Dieser Routineldentifizier wird entsprechend Dokument /1/ und /8/ verwendet, um alle Codierungen, Fahrzeug-, Kunden-, Werkstatt-, Prozessparameter, Erstbedatungswerte und Lernwerte (auch SFD-Ende-zu-Ende abgesicherte Konfigurationsdaten) auf die Werkseinstellungen (Ausgangszustand des Lieferanten) zurückzusetzen.

Hinweis: Ein Ausführen der Routine führt zu einer Änderung der IVD-relevanten Anteile des Konfigurations-Hashwertes wie Codierungen, Fahrzeugparameter und Erstbedatungswerte. Nach Ausführen dieser Routine ist es zwingend erforderlich eine Neubedatung mit ZDC durchzuführen.

[allg. Anf.: F-LAH\_RxSWIN-967]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Folgende Änderung gegenüber Q-LAH 80125 V5.3 bis V5.5 ist zu beachten:

- SFD-Zugriffsschutz (Rolle Basic) für Routineldentifizier 0x03E7.

## 5.2.2 Berechnung Integrity Validation Data

### 5.2.2.1 0x0253-Calculate\_integrity\_validation\_data

[I: F-LAH\_RxSWIN-451]

Dieser Routineldentifizier wird zur Berechnung der Integrity Validation Data von DK3/DK3V/DK4/DK4V/SWCL für Instruction-Code und Konfigurationsdaten verwendet.

[allg. Anf.: F-LAH\_RxSWIN-473]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Dieser Routineldentifizier muss mit dem hier geforderten Request-/Response-Verhalten und den hier geforderten Parametern umgesetzt werden. Er liefert mit der pos. Response auf den RoutineControlType 0x01 sein Ergebnis und wird automatisch beendet. Die Verwendung der Subfunctions

- StopRoutine (0x02)
- RequestRoutineResults (0x03)

ist für diesen Routineldentifizier nicht zulässig.

[allg. Anf.: F-LAH\_RxSWIN-597]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Das hier beschriebene Responseverhalten entspricht dem Q-LAH 80124 V2.x. Steuergeräte in Bestandsarchitekturen, die eine Version des Q-LAH 80124 1.x umgesetzt haben, müssen für den Routineldentifizier "0x0253-Calculate\_integrity\_validation\_data" die hier geforderte Response umsetzen. Bestehende Routinen auf Basis Q-LAH 80124 V1.x müssen im Steuergerät nicht angepasst werden.

[allg. Anf.: F-LAH\_RxSWIN-965]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Dieser Routineldentifizier muss bei Fahrzeugstillstand und im Zustand "keine Fahrbereitschaft" ausführbar sein. In anderen Fahrzeugzuständen (z.B. aktive Regeleingriffe) können Steuergeräte-spezifische Randbedingungen gelten. Die Signale zur Fahrbereitschaft und Geschwindigkeit müssen architekturabhängig gemäß gültiger Datenfestlegung verwendet werden.

[allg. Anf.: F-LAH\_RxSWIN-616]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-14 Aufbau Routineldentifizier 0x0253-Calculate\_integrity\_validation\_data**

| RID            | 0x0253   |
|----------------|--|
| Bezeichnung    | Calculate_integrity_validation_data  |
| Beschreibung   | Dieser Routineldentifizier wird zur Berechnung der Integrity Validation Data für Instruction-Code und Konfigurationsdaten verwendet. |
| Convention     | Alle Server  |
| Diagnoseklasse | DK3/DK3V, DK4/DK4-low /DK4V-low , DK4-high/DK4V-high, SWCL   |
| Session        | APP: 0x01, 0x03, 0x40<br>BLF: NoImp  |
| SecurityLevel  | NoImp  |
| Changing       | APP  |
| Format         | siehe Request/Response-Definition  |
| Range          | siehe Request/Response-Definition  |
| Init           | kein Init im Server  |
| Beispiel       | siehe Request/Response-Definition  |
| eBZD           | M  |

### 5.2.2.1.1 Request Message Definition

[I: F-LAH\_RxSWIN-453]

Folgende Request Message ist zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-454]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-15 Request Message Definition**

| Data | Description                   |  | Cvt. | Value (Hex) |
|------|-------------------------------|--|------|-------------|
| #1   | Request SID                   | RoutineControl                                     | M    | 31          |
| #2   | RoutineControlType            | StartRoutine                                       | M    | 01          |
| #3   | RoutineIdentifier [Byte#1]    | Calculate_integrity_validation_data [Byte#1] (MSB) | M    | 02          |
| #4   | RoutineIdentifier [Byte#2]    | Calculate_integrity_validation_data [Byte#2]       | M    | 53          |
| #5   | RoutineControlOption [Byte#1] | Type_of_calculation                                | M    | 00-FF       |
| #6   | RoutineControlOption [Byte#2] | Type_of_hash_value                                 | M    | 00-FF       |

### 5.2.2.1.2 Request Message Parameter Definition

[I: F-LAH\_RxSWIN-457]

Folgende Parameter sind zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-458]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-16 Request Message Parameter Definition**

|   |
|---|
| <b>Definition</b>   |
| <b>Type_of_calculation</b><br>Dieser Parameter gibt den Umfang der Hashwertberechnung an.<br><br>0x00 = Die Hashwertberechnung erfolgt über die Konfigurationsdaten (Data). Die Informationen über die Identifier werden dem Data-Identifier 0x0250 entnommen.<br>0x01 = Die Hashwertberechnung erfolgt über die Programmierdaten von Applikations- und Bootloader-Software (Instruction-Code).<br>Andere Werte sind VOLKSWAGEN AG-reserved |
| <b>Type_of_hash_value</b><br>Dieser Parameter gibt das Verfahren der Hashwertberechnung an.<br><br>0x01 = SHA-256<br>Andere Werte sind VOLKSWAGEN AG-reserved   |

### 5.2.2.1.3 Positive Response Message Definition

[I: F-LAH\_RxSWIN-460]

Folgende Response Message ist zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-461]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-17 Positive Response Message Definition**

| Data | Description                |  | Cvt. | Value (Hex) |
|------|----------------------------|--|------|-------------|
| #1   | Response SID               | RoutineControl                                     | M    | 71          |
| #2   | RoutineControlType         | StartRoutine                                       | M    | 01          |
| #3   | RoutineIdentifier [Byte#1] | Calculate_integrity_validation_data [Byte#1] (MSB) | M    | 02          |
| #4   | RoutineIdentifier [Byte#2] | Calculate_integrity_validation_data [Byte#2]       | M    | 53          |
| #5   | RoutineStatusRecord#1      | Result_of_calculation                              | M    | 00-FF       |
| #6   | RoutineStatusRecord#2      | Type_of_hash_value                                 | M    | 00-FF       |
| #7   | RoutineStatusRecord#3      | Hash_value [Byte#1] (MSB)                          | C1   | 00-FF       |
| :    | :                          | :  | :    | :           |
| #6+n | RoutineStatusRecord#2+n    | Hash_value [Byte#n]                                | C1   | 00-FF       |

[allg. Anf.: F-LAH\_RxSWIN-462]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

C1 = Länge abhängig vom Parameter [Result\_of\_calculation]:

- 32 Byte = Bei erfolgreiche Berechnung ([Result\_of\_calculation] = 0x00) eines SHA256-Hashwertes (Type\_of\_hash\_value = 0x01)
- 0 Byte = Bei einem Fehler ([Result\_of\_calculation] != 0x00)



#### 5.2.2.1.4 Positive Response Message Parameter Definition

[I: F-LAH\_RxSWIN-464]

Folgende Parameter sind zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-465]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-18 Positive Response Message Parameter Definition**

| Definition  |
|---|
| <b>Result_of_calculation</b><br>0x00 = Calculation_successful<br>Die Berechnung des Hashwertes wurde erfolgreich durchgeführt.<br><br>0x01 = Calculation_failed<br>Die Berechnung des Hashwertes ist fehlgeschlagen. Der Wert 0x01 ist nur auszugeben, wenn die Werte 0x02 bis 0x04 nicht zutreffen<br><br>0x02 = Calculation_identifier_not_found<br>Mindestens ein Dataidentifier oder eine Datensatznummer aus dem Dataidentifier 0x0250 ist im Diagnose-Server nicht vorhanden. Der Wert ist nur auszugeben, wenn Type_of_calculation = 0x00<br><br>0x03 = Calculation_no_identifier_found<br>Der Dataidentifier 0x0250 enthält keine Dataidentifier oder Datensatznummern (Z.B. vorgehaltener ZDC). Der Wert ist nur auszugeben, wenn Type_of_calculation = 0x00<br><br>0x04 = Calculation_incomplete<br>Bei der Berechnung des Programming-Hash-Werte eines LUM Flashcontainers liegt kein vollständiges Prüfergebnis vor. Der Wert ist nur auszugeben, wenn Type_of_calculation = 0x01<br><br>0x05 - 0xFF = VOLKSWAGEN AG-reserved |
| <b>Type_of_hash_value</b><br><br>0x00 = VOLKSWAGEN AG-Reserved<br>0x01 = SHA-256<br>Die Berechnung des Hashwertes erfolgt nach dem SHA-256-Verfahren.<br><br>0x02 - 0xFF = VOLKSWAGEN AG-reserved   |
| <b>Hash_value</b><br>n Byte <0x00 - 0xFF><br>Berechneter Hashwert, bei SHA-256 entspricht n = 32.   |

### 5.2.2.2 0x0254-Calculate\_individual\_hash\_value

[I: F-LAH\_RxSWIN-868]

Dieser Routinidentifizier wird zum Berechnen und Auslesen der Einzel-Hashwerte von DK3/DK3V/DK4/DK4V-Systemen und SWCL für Konfigurationsdaten verwendet.

[I: F-LAH\_RxSWIN-1137]

Dieser Routinidentifizier muss nur von DK3/DK3V/DK4/DK4V-Systemen und SWCL umgesetzt werden, die über Konfigurationsdaten verfügen und den Konfigurations-Hashwert umsetzen.

[allg. Anf.: F-LAH\_RxSWIN-930]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Dieser Routinidentifizier muss mit dem hier geforderten Request-/Response-Verhalten und den hier geforderten Parametern umgesetzt werden. Die Verwendung der Subfunctions

- StopRoutine (0x02)
- RequestRoutineResults (0x03)

ist für diesen Routinidentifizier nicht zulässig. Er liefert mit der pos. Response auf den RoutineControlType 0x01 sein Ergebnis und wird automatisch beendet.

[allg. Anf.: F-LAH\_RxSWIN-929]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Das hier beschriebene Responseverhalten entspricht dem Q-LAH 80124 V2.x. Steuergeräte in Bestandsarchitekturen, die eine Version des Q-LAH 80124 V1.x umgesetzt haben, müssen für den Routinidentifizier 0x0254-Calculate\_individual\_hash\_value" die hier geforderte Response umsetzen. Bestehende Routinen auf Basis Q-LAH 80124 V1.x müssen im Steuergerät nicht angepasst werden.

[allg. Anf.: F-LAH\_RxSWIN-1019]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Dieser Routinidentifizier muss bei Fahrzeugstillstand und im Zustand "keine Fahrbereitschaft" ausführbar sein. In anderen Fahrzeugzuständen (z.B. aktive Regeleingriffe) können Steuergeräte-spezifische Randbedingungen gelten. Die Signale zur Fahrbereitschaft und Geschwindigkeit müssen architekturabhängig gemäß gültiger Datenfestlegung verwendet werden.

[allg. Anf.: F-LAH\_RxSWIN-871]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-19 Aufbau Routinidentifizier 0x0254-Calculate\_individual\_hash\_value**

|                |   |
|----------------|---|
| <b>RID</b>     | <b>0x0254</b>   |
| Bezeichnung    | Calculate individual hash value   |
| Beschreibung   | Dieser Routinidentifizier wird zum Berechnen und Auslesen von Einzel-Hashwerte der Konfigurationsdaten verwendet. |
| Convention     | ZDC   |
| Diagnoseklasse | DK3/DK3V, DK4/DK4-low/DK4V-low, DK4-high/DK4V-high, SWCL  |
| Session        | APP: 0x01, 0x03, 0x40<br>BLF: NoImp   |
| SecurityLevel  | NoImp   |
| Changing       | APP   |
| Format         | siehe Request/Response-Definition   |
| Range          | siehe Request/Response-Definition   |
| Init           | kein Init im Server   |
| Beispiel       | siehe Request/Response-Definition   |
| eBZD           | NoImp   |

### 5.2.2.2.1 Request Message Definition

[I: F-LAH\_RxSWIN-873]

Folgende Request Message ist zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-874]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-20 Request Message Definition**

| Data | Description                   |  | Cvt. | Value (Hex) |
|------|-------------------------------|--|------|-------------|
| #1   | Request SID                   | RoutineControl                                 | M    | 31          |
| #2   | RoutineControlType            | StartRoutine                                   | M    | 01          |
| #3   | RoutineIdentifier [Byte#1]    | Calculate_individual_hash_value [Byte#1] (MSB) | M    | 02          |
| #4   | RoutineIdentifier [Byte#2]    | Calculate_individual_hash_value [Byte#2]       | M    | 54          |
| #5   | RoutineControlOption [Byte#1] | Type_of_hash_value                             | M    | 00-FF       |
| #6   | RoutineControlOption [Byte#2] | Type_of_hash                                   | M    | 00-FF       |
| #7   | RoutineControlOption [Byte#3] | Individual_hash_value_id                       | M    | 00-FF       |
| #8   | RoutineControlOption [Byte#4] | Individual_hash_value_id                       | M    | 00-FF       |

### 5.2.2.2.2 Request Message Parameter Definition

[I: F-LAH\_RxSWIN-876]

Folgende Parameter sind zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-877]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-21 Request Message Parameter Definition**

| Definition  |
|---|
| <b>Type_of_hash_value</b><br><br>0x00 = VOLKSWAGEN AG-reserved<br>0x01 = SHA-256 Die Berechnung des Hashwertes erfolgt nach dem SHA-256-Verfahren.<br>0x02 - 0xFF = VOLKSWAGEN AG-reserved  |
| <b>Type_of_hash</b><br><br>0x00 = Einzel-Hashwert über alle Datensätze (bei Datensatzdownload Gen. 1)<br>0x01 = Einzel-Hashwert über alle Bootloader-Datensätze (bei Datensatzdownload Gen. 2)<br>0x02 = Einzel-Hashwert Applikations-Datensatz (entspricht der Applikations-Datensatz-Nummer Datensatzdownload Gen. 2)<br>0x03 = Einzel-Hashwert über alle Anpassungen/Codierungen gemäß DataIdentifier 0x0250<br>0x04 - 0xFE = VOLKSWAGEN AG-reserved<br>0xFF = alle Einzel-Hashwerte gemäß DataIdentifier 0x0250 |
| <b>Individual_hash_value_id</b><br>Dieser Parameter identifiziert Einzel-Hashwerte von Applikationsdatensätzen gemäß DataIdentifier 0x0250.<br><br>0x7200 - 0x72FF = wird nur verwendet bei [Type_of_hash] = 0x02<br>0x0000 = bei allen anderen Werte im Parameter [Type_of_hash]<br>Alle anderen Werte sind VOLKSWAGEN AG-reserved   |

### 5.2.2.2.3 Positive Response Message Definition

[! : F-LAH\_RxSWIN-879]

Folgende Response Message ist zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-880]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-22 Positive Response Message Definition**

| Data     | Description                |  | Cvt. | Value (Hex) |
|----------|----------------------------|--|------|-------------|
| #1       | Response SID               | RoutineControl                                 | M    | 71          |
| #2       | RoutineControlType         | StartRoutine                                   | M    | 01          |
| #3       | RoutineIdentifier [Byte#1] | Calculate_individual_hash_value [Byte#1] (MSB) | M    | 02          |
| #4       | RoutineIdentifier [Byte#2] | Calculate_individual_hash_value [Byte#2]       | M    | 54          |
| #5       | RoutineStatusRecord#1      | State_of_hash_value                            | M    | 00-FF       |
| #6       | RoutineStatusRecord#2      | Type_of_hash_value                             | M    | 00-FF       |
| #7       | RoutineStatusRecord#3      | Type_of_hash                                   | M    | 00-FF       |
| #8       | RoutineStatusRecord#4      | Individual_hash_value_id [Byte#1] (MSB)        | M    | 00-FF       |
| #9       | RoutineStatusRecord#5      | Individual_hash_value_id [Byte#2]              | M    | 00-FF       |
| #10      | RoutineStatusRecord#6      | Hash_value#1 [Byte#1] (MSB)                    | C1   | 00-FF       |
| :        | :                          | :  | :    | :           |
| #9+n     | RoutineStatusRecord#6+n    | Hash_value#1 [Byte#n]                          | C1   | 00-FF       |
| #10 + n  | RoutineStatusRecord#6+n+1  | Hash_value#m [Byte#1] (MSB)                    | C2   | 00-FF       |
| :        | :                          | :  | :    | :           |
| #9+n + p | RoutineStatusRecord#6+n+p  | Hash_value#m [Byte#p]                          | C2   | 00-FF       |

[allg. Anf.: F-LAH\_RxSWIN-881]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

C1 = Nur vorhanden wenn ein Einzel-Hashwert verfügbar ist.

[allg. Anf.: F-LAH\_RxSWIN-885]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

C2 = Nur vorhanden wenn der Parameter [Type\_of\_hash] den Wert 0xFF hat und mehr als ein Hashwert verfügbar ist.

#### 5.2.2.2.4 Positive Response Message Parameter Definition

[I: F-LAH\_RxSWIN-883]

Folgende Parameter sind zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-884]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-23 Positive Response Message Parameter Definition**

| Definition   |
|--|
| <b>State_of_hash_value</b><br>0x00 = Calculation_hash_value_valid<br>Die Berechnung des Hashwertes wurde erfolgreich durchgeführt und der Hashwert ist gültig.<br><br>0x01 = Calculation_hash_value_invalid<br>Die Berechnung des Hashwertes ist fehlgeschlagen und der Hashwert ist nicht gültig.<br><br>0x02 = Calculation_hash_value_not_found<br>Der Identifier für den Hashwert ist nicht gültig und im Dataidentifier 0x0250 im Diagnose-Server nicht vorhanden.<br><br>0x03 = Calculation_no_identifier_found<br>Der Dataidentifier 0x0250 enthält keine Dataidentifier oder Datensatznummern (Z.B. vorgehaltener ZDC).<br><br>0x04 - 0xFF = VOLKSWAGEN AG-reserved |
| <b>Type_of_hash_value</b><br>0x00 = VOLKSWAGEN AG-Reserved<br>0x01 = SHA-256, Die Berechnung des Hashwertes erfolgt nach dem SHA-256-Verfahren.<br>0x02 - 0xFF = VOLKSWAGEN AG-reserved  |
| <b>Type_of_hash</b><br>0x00 = Einzel-Hashwert über alle Datensätze (bei Datensatzdownload Gen. 1) Hinweis: DSDL Gen1 wird ab UNECE V1.3 nicht mehr unterstützt<br>0x01 = Einzel-Hashwert über alle Bootloader-Datensätze (bei Datensatzdownload Gen. 2)<br>0x02 = Einzel-Hashwert Applikations-Datensatz (entspricht der Applikations-Datensatz-Nummer Datensatzdownload Gen. 2)<br>0x03 = Einzel-Hashwert über alle Anpassungen/Codierungen gemäß Dataidentifier 0x0250<br>0x04 - 0xFE = VOLKSWAGEN AG-reserved<br>0xFF = alle Einzel-Hashwerte gemäß Dataidentifier 0x0250   |
| <b>Individual_hash_value_id</b><br>Dieser Parameter identifiziert Einzel-Hashwerte von Applikationsdatensätzen gemäß Dataidentifier 0x0250.<br><br>0x7200 - 0x72FF = wird nur verwendet bei [Type_of_hash] = 0x02<br>0x0000 = bei allen anderen Werte im Parameter [Type_of_hash]<br>Alle anderen Werte sind VOLKSWAGEN AG-reserved  |
| <b>Hash_value</b><br>Bei Type_of_hash: 0x00 .. 0x03<br><n Byte <0x00 - 0xFF>>,<br>Berechneter Hashwert,<br>bei SHA-256 entspricht n = 32,<br><br>Bei Type_of_hash: 0xFF<br><m-mal <n Byte <0x00 - 0xFF>>,<br>Berechneter Hashwert,<br>bei SHA-256 entspricht n = 32,<br>(m: Anzahl aller individuellen Hashwerte gemäß Dataidentifier 0x0250)  |

[allg. Anf.: F-LAH\_RxSWIN-1136]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Wenn die Routine 0x0254 mit der RoutineControlOption "Type\_of\_hash" = 0xFF = "alle Einzel-Hashwerte gemäß Dataidentifier 0x0250-Integrity\_validation\_data\_configuration\_list" angefragt wird, muss der positive Response der Routine 0x0254 die Einzel-Hashwerte in der Reihenfolge ausgeben, die durch die Gruppierung der Datensatznummern und Dataidentifier und der Datensatznummern bzw. Dataidentifiern im Dataidentifier "0x0250-Integrity\_validation\_data\_configuration\_list" vorgeben werden.

### 5.2.2.3 0x029A-Calculate\_module\_hash\_value

[I: F-LAH\_RxSWIN-1021]

Für nonEmbedded bzw. filebasierte Systeme sowie Embedded-Systeme der Diagnoseklassen DK2F/DK2FV/DK3/DK3V/DK4/DK4V/SWCL mit systemspezifischer IVD-Berechnungsmethode (d. h. der Update-Container (LUM/ODX-F) enthält den Soll-Hashwert der Programmierung) wird dieser Routinidentifizier zur vollständigen Neuberechnung der Integrity Validation Data für den Instruction-Code verwendet.

[allg. Anf.: F-LAH\_RxSWIN-1022]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Dieser Routinidentifizier muss mit dem hier geforderten Request-/Response-Verhalten und den hier geforderten Parametern umgesetzt werden. Er liefert mit der positiven Response auf den RoutineControlType 0x01 noch kein endgültiges Ergebnis, sondern startet die vollständige Neuberechnung des Programmierungs-Hashwertes. Diese Routine wird nicht automatisch beendet. Die Verwendung der Subfunctions

- StopRoutine (0x02)
- RequestRoutineResults (0x03)

ist für diesen Routinidentifizier verbindlich umzusetzen.

[allg. Anf.: F-LAH\_RxSWIN-1023]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Das hier beschriebene Responseverhalten entspricht dem Q-LAH 80124, V2.x. Steuergeräte in Bestandsarchitekturen, die eine Version des Q-LAH 80124 1.x umgesetzt haben, müssen für den Routinidentifizier "0x029A-Calculate\_module\_hash\_value" die hier geforderte Response umsetzen. Bestehende Routinen auf Basis Q-LAH 80124 V1.x müssen im Steuergerät nicht angepasst werden.

[allg. Anf.: F-LAH\_RxSWIN-1024]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

Dieser Routinidentifizier muss bei Fahrzeugstillstand und im Zustand "keine Fahrbereitschaft" ausführbar sein. In anderen Fahrzeugzuständen (z.B. aktive Regeleingriffe) können Steuergeräte-spezifische Randbedingungen gelten. Die Signale zur Fahrbereitschaft und Geschwindigkeit müssen architekturabhängig gemäß gültiger Datenfestlegung verwendet werden.

[allg. Anf.: F-LAH\_RxSWIN-1025]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-24 Aufbau Routinidentifizier 0x029A-Calculate\_module\_hash\_value**

|                       |   |
|-----------------------|---|
| <b>RID</b>            | <b>0x029A</b>   |
| <b>Bezeichnung</b>    | Calculate_module_hash_value   |
| <b>Beschreibung</b>   | Dieser Routinidentifizier wird zur vollständigen Neuberechnung der Integrity Validation Data für Instruction-Code im Fall von file-basierten (non-embedded) und embedded Systemen verwendet, in deren Update-Container (LUM/ODX-F) der Soll-Programming Hashwert enthalten ist. |
| <b>Convention</b>     | Alle Server   |
| <b>Diagnoseklasse</b> | DK2F/DK3/DK3V, DK4/DK4-low/DK4V-low, DK4-high/DK4V-high, SWCL   |
| <b>Session</b>        | DK3/DK3V, DK4/DK4-low/DK4V-low, DK4-high/DK4V-high, SWCL:<br>APP: 0x01, 0x03, 0x40<br>DK2F/DK2FV:<br>APP: 0x01 (U), 0x03 (U), 0x40 (U)<br>BLF: 0x01, 0x02, 0x03   |
| <b>SecurityLevel</b>  | NoImp   |
| <b>Changing</b>       | APP   |
| <b>Format</b>         | siehe Request/Response-Definition   |
| <b>Range</b>          | siehe Request/Response-Definition   |
| <b>Init</b>           | kein Init im Server   |
| <b>Beispiel</b>       | siehe Request/Response-Definition   |
| <b>eBZD</b>           | NoImp   |

### 5.2.2.3.1 StartRoutine (0x01)

#### 5.2.2.3.1.1 Request Message Definition

[I: F-LAH\_RxSWIN-1028]

Folgende Request Message ist zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-1029]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-25 Request Message Definition**

| Data | Description                   | Cvt. | Value (Hex) |
|------|-------------------------------|------|-------------|
| #1   | Request SID                   | M    | 31          |
| #2   | RoutineControlType            | M    | 01          |
| #3   | RoutineIdentifier [Byte#1]    | M    | 02          |
| #4   | RoutineIdentifier [Byte#2]    | M    | 9A          |
| #5   | RoutineControlOption [Byte#1] | M    | 00-FF       |
| #6   | RoutineControlOption [Byte#2] | M    | 00-FF       |

#### 5.2.2.3.1.2 Request Message Parameter Definition

[I: F-LAH\_RxSWIN-1032]

Folgende Parameter sind zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-1033]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-26 Request Message Parameter Definition**

| Definition   |
|--|
| <b>Type_of_hash_value</b><br>Dieser Parameter gibt das Verfahren der Hashwertberechnung an.<br><br>0x01 = SHA-256<br>Andere Werte sind VOLKSWAGEN AG reserved  |
| <b>Type_of_calculation</b><br>Dieser Parameter gibt den Umfang der Hashwertberechnung an.<br><br>0x01 = Die Hashwertberechnung erfolgt über die Programmierdaten zur vollständigen Neuberechnung des Programmierungs-Hashwertes.<br>Andere Werte sind VOLKSWAGEN AG reserved |

#### 5.2.2.3.1.3 Positive Response Message Definition

[I: F-LAH\_RxSWIN-1035]

Folgende Response Message ist zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-1036]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-27 Positive Response Message Definition**

| Data | Description                | Cvt. | Value (Hex) |
|------|----------------------------|------|-------------|
| #1   | Response SID               | M    | 71          |
| #2   | RoutineControlType         | M    | 01          |
| #3   | RoutineIdentifier [Byte#1] | M    | 02          |
| #4   | RoutineIdentifier [Byte#2] | M    | 9A          |
| #5   | RoutineInfo                | M    | XX          |
| #6   | RoutineStatusRecord#1      | M    | 00-FF       |
| #7   | RoutineStatusRecord#2      | M    | 00-FF       |

#### 5.2.2.3.1.4 Positive Response Message Parameter Definition

[I: F-LAH\_RxSWIN-1038]

Folgende Parameter sind zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-1039]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

#### Tabelle 5-28 Response Message Parameter Definition

| Definition   |  |
|--|--|
| <b>Type_of_hash_value</b>  | Dieser Parameter gibt das Verfahren der Hashwertberechnung an. |
| 0x01 = SHA-256   |  |
| Andere Werte sind VOLKSWAGEN AG reserved   |  |
| <b>Type_of_calculation</b>   | Dieser Parameter gibt den Umfang der Hashwertberechnung an.    |
| 0x01 = Die Hashwertberechnung erfolgt über die Programmierdaten zur vollständigen Neuberechnung des Programmierungshashwertes. |  |
| Andere Werte sind VOLKSWAGEN AG reserved   |  |

#### 5.2.2.3.2 StopRoutine (0x02)

##### 5.2.2.3.2.1 Request Message Definition

[I: F-LAH\_RxSWIN-1041]

Folgende Request Message ist zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-1042]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

#### Tabelle 5-29 Request Message Definition

| Data | Description                |  | Cvt. | Value (Hex) |
|------|----------------------------|--|------|-------------|
| #1   | Request SID                | RoutineControl                             | M    | 31          |
| #2   | RoutineControlType         | StopRoutine                                | M    | 02          |
| #3   | RoutineIdentifier [Byte#1] | Calculate module hash value [Byte#1] (MSB) | M    | 02          |
| #4   | RoutineIdentifier [Byte#2] | Calculate module hash value [Byte#2]       | M    | 9A          |

##### 5.2.2.3.2.2 Positive Response Message Definition

[I: F-LAH\_RxSWIN-1044]

Folgende Response Message ist zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-1045]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

#### Tabelle 5-30 Positive Response Message Definition

| Data | Description                |  | Cvt. | Value (Hex) |
|------|----------------------------|--|------|-------------|
| #1   | Response SID               | RoutineControl                             | M    | 71          |
| #2   | RoutineControlType         | StopRoutine                                | M    | 02          |
| #3   | RoutineIdentifier [Byte#1] | Calculate module hash value [Byte#1] (MSB) | M    | 02          |
| #4   | RoutineIdentifier [Byte#2] | Calculate module hash value [Byte#2]       | M    | 9A          |



### 5.2.2.3.3 Request Routine Result (0x03)

#### 5.2.2.3.3.1 Request Message Definition

[! : F-LAH\_RxSWIN-1047]

Folgende Request Message ist zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-1048]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-31 Request Message Definition**

| Data | Description                | Cvt. | Value (Hex) |
|------|----------------------------|------|-------------|
| #1   | Request SID                | M    | 31          |
| #2   | RoutineControlType         | M    | 03          |
| #3   | RoutineIdentifier [Byte#1] | M    | 02          |
| #4   | RoutineIdentifier [Byte#2] | M    | 9A          |

#### 5.2.2.3.3.2 Positive Response Message Definition

[! : F-LAH\_RxSWIN-1054]

Folgende Response Message ist zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-1055]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-32 Positive Response Message Definition**

| Data    | Description                | Cvt. | Value (Hex) |
|---------|----------------------------|------|-------------|
| #1      | Response SID               | M    | 71          |
| #2      | RoutineControlType         | M    | 03          |
| #3      | RoutineIdentifier [Byte#1] | M    | 02          |
| #4      | RoutineIdentifier [Byte#2] | M    | 9A          |
| #5      | RoutineInfo                | M    | XX          |
| #6      | RoutineStatusRecord#1      | M    | 00-64; FF   |
| #7      | RoutineStatusRecord#2      | M    | 00-FF       |
| #8      | RoutineStatusRecord#3      | M    | 00-FF       |
| #9      | RoutineStatusRecord#4      | M    | 00-FF       |
| #10     | RoutineStatusRecord#5      | C1   | 00-FF       |
| :       | :                          | :    | :           |
| #10 + n | RoutineStatusRecord#5 + n  | C1   | 00-FF       |

[allg. Anf.: F-LAH\_RxSWIN-1057]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

C1 = Länge abhängig vom Parameter [Result\_of\_calculation]:

32 Byte = Bei erfolgreicher Berechnung ([Result\_of\_calculation] = 0x00) eines SHA256-Hashwertes (Type\_of\_hash\_value = 0x01)

0 Byte = Bei einem Fehler ([Result\_of\_calculation] != 0x00)

### 5.2.2.3.3 Response Message Parameter Definition

[I: F-LAH\_RxSWIN-1051]

Folgende Parameter sind zu implementieren:

[allg. Anf.: F-LAH\_RxSWIN-1052]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

**Tabelle 5-33 Response Message Parameter Definition**

| Definition   |
|--|
| <p><b>Progress_of_calculation</b><br/>Fortschritt der Berechnung des aktuell bearbeiteten Installationspaketes (Blocks).<br/>0 ... 100% = Fortschritt des Updates<br/>0xFF = keine Information vorhanden</p> <p>1. Für Result_of_calculation = Calculation_successful<br/>Progress_of_calculation = 0x64 (100%)<br/>2. Für Result_of_calculation = Calculation_failed<br/>Progress_of_calculation = 0x00 (0%)<br/>3. Für Calculation_incomplete<br/>Progress_of_calculation = 0x00... 0x64 (0..100%)<br/>4. Für Result_of_calculation = 0x02 - 0x03 und 0x05 - 0xFF<br/>Progress_of_calculation = 0xFF</p> |
| <p><b>Type_of_hash_value</b><br/>Dieser Parameter gibt das Verfahren der Hashwertberechnung an.</p> <p>0x01 = SHA-256<br/>Andere Werte sind VOLKSWAGEN AG reserved</p>   |
| <p><b>Type_of_calculation</b><br/>Dieser Parameter gibt den Umfang der Hashwertberechnung an.</p> <p>0x01 = Die Hashwertberechnung erfolgt über die Programmierdaten zur vollständigen Neuberechnung des Programmierungs-Hashwertes.<br/>Andere Werte sind VOLKSWAGEN AG reserved</p>  |
| <p><b>Result_of_calculation</b><br/>0x00 = Calculation_successful<br/>Die Berechnung des Hashwertes wurde erfolgreich durchgeführt.</p> <p>0x01 = Calculation_failed<br/>Die Berechnung des Hashwertes ist fehlgeschlagen.<br/>Hinweis: Der Wert 0x01 ist nur auszugeben, wenn die Werte 0x02 bis 0x05 nicht zutreffen.</p> <p>0x02 - 0x03 = VOLKSWAGEN AG-reserved</p> <p>0x04 = Calculation_incomplete<br/>Bei der Berechnung des Programming-Hashwerte eines LUM-Flashcontainers liegt kein vollständiges Prüfergebnis vor.</p> <p>0x05 - 0xFF = VOLKSWAGEN AG-reserved</p>                             |
| <p><b>Hash_value</b><br/>n Byte &lt;0x00 - 0xFF&gt;<br/>Berechneter Hashwert bei SHA-256 entspricht n = 32</p>   |

## 6 RxSWIN-spezifische Dokumentation

[I: F-LAH\_RxSWIN-55]

Zu einer RxSWIN können mehrere "Steuergeräte-Versionen" mit ihren Identifikationsdaten und Integrity Validation Data zugeordnet sein. Die nachfolgenden Daten müssen mindestens zu jeder RxSWIN in den IT-Systemen dokumentiert werden.

### 6.1 Daten für die RxSWIN-spezifische Dokumentation eines DK3/DK3V/DK4/DK4V-System

[Prozess-Anf.: F-LAH\_RxSWIN-149]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- Nur Gateway-Steuergeräte: Fahrgestellnummer (Vehicle Identification Number) aus DataIdentifier 0xF190
- Nur Gateway-Steuergeräte: Liste der RxSWINs aus dem DataIdentifier 0xF18F
- VW/Audi-Teilnummer (VW Spare Part Number) aus DataIdentifier 0xF187
- Softwareversion (VW Application Software Version Number) aus DataIdentifier 0xF189
- Hardwareteilnummer (VW ECU Hardware Number) aus DataIdentifier 0xF191
- optional: Hardwareversion (VW ECU Hardware Version Number) aus DataIdentifier 0xF1A3
- FAZIT-Identifikation (FAZIT-Identification String) aus DataIdentifier 0xF17C
- Identifikation des ZDCs (Teilnummer und Version des ZDCs)
- Integrity Validation Data der Programmierung (Programming\_hash) aus RoutineIdentifier 0x0253
- Integrity Validation Data der Konfiguration (Configuration\_hash) aus RoutineIdentifier 0x0253
- Diagnoseadresse (DA)

## 6.2 Daten für die RxSWIN-spezifische Dokumentation eines SWCL

[Prozess-Anf.: F-LAH\_RxSWIN-428]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- VW/Audi-Teilnummer (VW Spare Part Number) aus DataIdentifier 0xF187
- Softwareversion (VW Application Software Version Number) aus DataIdentifier 0xF189
- Identifikation des ZDCs (Teilnummer und Version des ZDCs)
- Integrity Validation Data der Programmierung (Programming\_hash) aus RoutineIdentifier 0x0253
- Integrity Validation Data der Konfiguration (Configuration\_hash) aus RoutineIdentifier 0x0253
- Diagnoseadresse (DA)

### 6.3 Daten für die RxSWIN-spezifische Dokumentation eines DK2/DK2F/DK2FV-System

[Prozess-Anf.: F-LAH\_RxSWIN-89]

[BsM-E.D: nein, BsM-E.L: nein, BsM-E.T: nein, BsM-O: nein, BsM-Sa.FuSi: nein]

- VW/Audi-Teilnummer (VW Spare Part Number) aus DataIdentifier 0xF187 bzw. 0x6200..63FF
- Hardwareteilnummer (VW ECU Hardware Number) aus DataIdentifier 0xF191 bzw. 6600..67FF
- Softwareversion (VW Application Software Version Number) aus DataIdentifier 0xF189 bzw. 0x6400..65FF
- optional: Hardwareversion (VW ECU Hardware Version Number) aus DataIdentifier 0xF1A3 bzw. 6800..69FF
- optional: FAZIT-Identifikation (FAZIT-Identification String) aus DataIdentifier 0xF17C bzw. 6E00..6FFF
- nur bei DK2F/DK2FV: Integrity Validation Data der Programmierung (Programming\_hash) aus DataIdentifier 0x0249 bzw. 0xA800..A9FF
- Identifikation des ZDCs (Teilnummer und Version des ZDCs)
- Integrity Validation Data der Konfiguration (Configuration\_hash) aus DataIdentifier 0x0245 bzw. AA00..ABFF
- Diagnoseadresse (DA)

## 7 Mitgeltende Dokumente und Spezifikationen

- [I: F-LAH\_RxSWIN-5]  
/1/ LAH.DUM.909.G - Q-LAH 80124 - Unified Diagnostic Services Protocol (UDS), Emissions-related Diagnostic Services (legislated OBD)
- [I: F-LAH\_RxSWIN-48]  
/2/ LAH.DUM.909.H - Q-LAH 80125 - Identifikation elektronischer Fahrzeugsysteme
- [I: F-LAH\_RxSWIN-49]  
/3/ LAH.DUM.909.B - Q-LAH 80127 - Diagnose verteilter Systeme; Diagnoseanforderungen an Bus-master und Systeme
- [I: F-LAH\_RxSWIN-69]  
/4/ LAH.DUM.907.BD - Q-LAH Schutz der Fahrzeugdiagnose (SFD)
- [I: F-LAH\_RxSWIN-75]  
/5/ LAH.DUM.000.AD - Flashdatensicherheit für UDS Steuergeräte
- [I: F-LAH\_RxSWIN-76]  
/6/ LAH.DUM.906.A - Q-LAH 80126 - UDS konforme Programmierung von Steuergeräten
- [I: F-LAH\_RxSWIN-144]  
/7/ LAH.DUM.907.R1 - Datensatzdownload Generation 2
- [I: F-LAH\_RxSWIN-242]  
/8/ LAH.DUM.907.BE - QLAH\_Datenarten
- [I: F-LAH\_RxSWIN-540]  
/9/ LAH Diagnosefilter
- [I: F-LAH\_RxSWIN-543]  
/10/ Entfällt - Keine Beschreibung
- [I: F-LAH\_RxSWIN-596]  
/11/ Datensatzdownload Generation 1
- [I: F-LAH\_RxSWIN-648]  
/12/ LAH.000.036.G - Q-LAH 80127ES - Ergänzungsspezifikation für hochintegrierte Systeme
- [I: F-LAH\_RxSWIN-649]  
/13/ LAH.000.036.H - Update filebasierter Systeme - Ergänzungsspezifikation für Non-Embedded-Systeme
- [I: F-LAH\_RxSWIN-650]  
/14/ LAH.DUM.906.B - Q-LAH 80128 Teil 3 - Spezifikation für Flashcontainer ODX-Flash PDX-Flash
- [I: F-LAH\_RxSWIN-697]  
/15/ LAH.000.900.AT - VKMS-Kernfunktionalität
- [I: F-LAH\_RxSWIN-702]  
/16/ LAH.DUM.907.Q1 - Diagnoseschnittstelle SWaP
- [I: F-LAH\_RxSWIN-703]  
/17/ LAH.000.900.AQ/BA - Secure activation of car functions - FoD-Master/FoD-Slave
- [I: F-LAH\_RxSWIN-704]  
/18/ LAH.DUM.907.xx - Lastenheft Wegfahrsperre Master/Slave
- [I: F-LAH\_RxSWIN-705]  
/19/ LAH.DUM.907.xx - Lastenheft Komponentenschutz Master/Slave
- [I: F-LAH\_RxSWIN-706]  
/20/ LAH.DUM.900.AB - Hardware Security Module
- [I: F-LAH\_RxSWIN-1117]  
/21/ ECU Manifest (Train release) specification

[I: F-LAH\_RxSWIN-1143]

/22/ LAH.893.910.A - Standardsoftware