# Feature – Backup Start Passcode
# a.k.a. PaaK Backup

# Requirements Specification

Version 2.7
Version Date: June 8, 2018

# Revision History

| Date | Version | Notes |
|---|---|---|
| May 8, 2017 | 2.1 | Initial Release |
| June 23, 2017 | 2.2 | Second Release |
| June 30, 2017 | 2.3 | Third Release |
| Sept 22, 2017 | 2.4 | Fourth Release |
| | | adelong2: Corrected signal name for signal 2 – the password entry display signal. |
| | | adelong2: Updated system diagram (section 1.3) to include correct vehicle architecture (CGEA1.3C). |
| | | adelong2: LBI.R001.03 – Updated wording. Changed number of times for opt-in message to appear in SYNC from one to five. |
| | | adelong2: LBI.R003.03 – Changed what happens after user selects option to create password; they will see a screen with requirements and steps. |
| | | adelong2: LBI.R330.01 – New. Split from R003. |
| | | adelong2: LBI.R183.02 – Added “Interior” to Registry search. Added CAN signals for specifying target zone and for encryption data. Added note. |
| | | adelong2: LBI.R184.02 – Added “Interior” to Registry search. Added CAN signals for specifying target zone and for encryption data. |
| | | adelong2: LBI.R006.02 – Added “interior” to Registry search. Added note. |
| | | adelong2: LBI.R011.03 – Updated wording. |
| | | adelong2: LBI.R012.03 – Added “Interior” to Registry search. Added CAN signals for specifying target zone and for encryption data. |
| | | adelong2: LBI.R185.02 – Added “Interior” to Registry search. Added CAN signals for specifying target zone and for encryption data. |
| | | adelong2: LBI.R013.02 – Added “interior” to Registry search. |
| | | adelong2: LBI.R331.01 – New. Addresses storage failure. |
| | | adelong2: LBI.R332.01 – New. Addresses storage failure. |
| | | adelong2: LBI.R017.02 – Changed HMI behavior. |
| | | adelong2: LBI.R022.03 – Added “up to” to programming mode time. |
| | | adelong2: LBI.R025.03 – Added requirement for BLEM to keep track of whether a device/key index has an associated keypad code, which is used for determining when to send requests to BCM. |
| | | adelong2: LBI.R333.01 – New. Added requirement here to reflect desired HMI behavior. |
| | | adelong2: Updated use case for creating backup password. Changed HMI behavior in SYNC. |
| | | adelong2: Updated sequence diagrams for creating backup passwords. Added missing CAN signals. |
| | | adelong2: LBI.R027.04 – Removed “restricted” from Infotainment Mode. There will be no infotainment restrictions for the password entry screen. |
| | | adelong2: LBI.R305.01 – Deleted. No restrictions. |
| | | adelong2: Note and R195, R196 and R306 moved to General Requirements for SYNC, section 3.1.3. |
| | | adelong2: LBI.R196.03 – Updated logic for what happens after 30 seconds of inactivity. What happens now also depends on whether delay accessory or extended play mode is active. |
| | | adelong2: LBI.R306.02 – Updated logic for what happens after lockout pop-up expires. What happens now also depends on whether delay accessory or extended play mode is active. |

| | | |
|---|---|---|
| | | adelong2: LBI.R197.02 – Added “backup” to the requirement. |
| | | adelong2: LBI.R201.03 – Updated wording. Removed reference. |
| | | adelong2: LBi.R202.03 – Split requirement. |
| | | adelong2: LBI.R334.01 – New. Split from R202. Noted all scenarios in which lockout applies i.e. all scenarios when a password is verified. Updated wording. |
| | | adelong2: LBI.R041.02 – Updated wording. Removed reference. |
| | | adelong2: LBI.R203.02 – Added “Interior” to Registry search. Added CAN signals for specifying target zone and for encryption data. |
| | | adelong2: LBI.R204.02 – Added “Interior” to Registry search. Added CAN signals for specifying target zone and for encryption data. |
| | | adelong2: LBI.R205.02 – Added “Interior” to Registry search. |
| | | adelong2: LBI.R045.02 – Corrected typo. |
| | | adelong2: LBI.R047.03 – Added “hash” to password. |
| | | adelong2: LBI.R048.03 – Added “hash” to password. Updated wording. |
| | | adelong2: LBI.R207.02 – Added “hash” to password. Included logic for determining when to send delete requests to BCM based on keypad status. |
| | | adelong2: LBI.R335.01 – New. Addresses deletion failure. |
| | | adelong2: LBI.R336.01 – New. Addresses deletion failure. |
| | | adelong2: LBI.R049.03 – Added “up to” to programming mode time. |
| | | adelong2: LBI.R310.02 – Updated to new TP RspStatus. |
| | | adelong2: LBI.R311.02 – Updated to new TP RspStatus. Changed HMI behavior in SYNC. |
| | | adelong2: Updated sequence diagram for deleting backup passwords. Added missing CAN signals. |
| | | adelong2: LBI.R210.02 – Added “Interior” to Registry search. Added CAN signals for specifying target zone and for encryption data. |
| | | adelong2: LBI.R211.02 – Added “Interior” to Registry search. Added CAN signals for specifying target zone and for encryption data. |
| | | adelong2: LBI.R054.03 – Added “Interior” to Registry search. |
| | | adelong2: LBI.R212.02 – Added “backup” to the requirement. |
| | | adelong2: LBI.R213.02 – Added “backup” to the requirement. |
| | | adelong2: LBI.R062.03 – Included logic for determining when to send delete requests to BCM based on keypad status. Removed requirement to delete backup password before deleting keypad code for resetting function only. |
| | | adelong2: LBI.R063.03 – Added “up to” to programming mode time. |
| | | adelong2: LBI.R066.03 – Updated requirement to reflect change that backup password will be deleted after confirmation of associated keypad code deletion (if applicable). This applies to reset function only. |
| | | adelong2: LBI.R337.01 – Updated requirement to reflect that new backup password is stored after current one is deleted. |
| | | adelong2: LBI.R338.01 – New. Addresses deletion/reset failure. |
| | | adelong2: LBI.R339.01 – New. Addresses deletion/reset failure. |
| | | adelong2: LBI.R340.01 – New. Addresses keypad code deletion failure. |
| | | adelong2: LBI.R341.01 – New. Addresses keypad code deletion failure. |
| | | adelong2: LBI.R342.01 – New. Addresses keypad code deletion failure. |
| | | adelong2: LBI.R343.01 – New. Describes HMI behavior for reset completion. |
| | | adelong2: Updated sequence diagrams for resetting backup passwords. Added missing CAN signals. Corrected typo. Changed to delete password after keypad code. |
| | | adelong2: LBI.R077.03 – Updated wording. Combined R234 into this requirement. |

| | | |
|---|---|---|
| | | adelong2: LBI.R234.01 – Deleted. Combined content with R077. |
| | | adelong2: LBI.R078.03 – Updated logic. Updated wording. Removed “with passwords” from device search requirements. PaaK devices are not required to have an associated backup password to be used for authenticating Enhanced Valet Mode. |
| | | adelong2: LBI.R235.02 – Added “Interior” to Registry search. Added CAN signals for specifying target zone and for encryption data. |
| | | adelong2: LBI.R236.02 – Added “Interior” to Registry search. Added CAN signals for specifying target zone and for encryption data. |
| | | adelong2: LBI.R237.02 – Added “Interior” to Registry search. Updated to new TP RspStatuses. Removed “with passwords” from device search requirements. |
| | | adelong2: LBI.R079.03 – Updated to new TP RspStatus. |
| | | adelong2: LBI.R238.02 – Removed “with passwords” from device search requirements. |
| | | adelong2: LBI.R081.03 – Updated wording. |
| | | adelong2: LBI.R082.03 – Added “up to” to programming mode time. |
| | | adelong2: LBI.R241.02 – Corrected error regarding source. |
| | | adelong2: LBI.R242.02 – Updated wording. Defined payload for delivering valet password to mobile device. |
| | | adelong2: LBI.R086.03 – Updated logic. Defined payload for confirming valet password delivery from mobile device. Updated to new TP RspStatus. |
| | | adelong2: LBI.R344.01 – New. Addresses valet password delivery failure. |
| | | adelong2: LBI.R345.01 – New. Addresses HMI behavior when password delivery is successful. |
| | | adelong2: LBI.R087.03 – Updated logic to reflect unsuccessful password delivery. |
| | | adelong2: LBI.R088.03 – Updated to new TP RspStatus. |
| | | adelong2: LBI.R244.02 – Added “backup” to the requirement. |
| | | adelong2: LBI.R097.03 – Updated wording. |
| | | adelong2: LBI.R248.03 – Corrected error regarding source. |
| | | adelong2: LBI.R346.01 – New. Defines infotainment restrictions for Enhanced Valet Mode. |
| | | adelong2: LBI.R250.02 – Updated wording. |
| | | adelong2: Updated use case for generating valet password. Removed “with password” from pre-condition about PaaK devices in vehicle. |
| | | adelong2: Updated sequence diagrams for generating valet passwords. Added missing CAN signals. |
| | | adelong2: LBI.R103.04 – Updated wording. Removed “restricted” from Infotainment Mode. |
| | | adelong2: LBI.R347.01 – LBI.R351.01 – New requirements for this function, but essentially duplicates from starting the vehicle function. |
| | | adelong2: LBI.R251.03 – Updated to new TP OpCode. |
| | | adelong2: LBI.R352.01 – LBI.R354.01 – New requirements for this function, but essentially duplicates from starting the vehicle function. |
| | | adelong2: Updated use case for starting vehicle with valet password. Added pre-condition. |
| | | adelong2: LBI.R253.02 – Added “Interior” to Registry search. Added CAN signals for specifying target zone and for encryption data. |
| | | adelong2: LBI.R254.02 – Added “Interior” to Registry search. Added CAN signals for specifying target zone and for encryption data. |

| | | |
|---|---|---|
| | | adelong2: LBI.R105.03 – Added “Interior” to Registry search. Added “hash” to password. Added missing reference requirement. Updated to new TP RspStatus |
| | | adelong2: LBI.R355.01 – New. Addresses valet password deletion failure. |
| | | adelong2: LBI.R356.01 – New. Addresses valet password deletion failure. |
| | | adelong2: LBI.R255.02 – Corrected error regarding source. Added note. |
| | | adelong2: LBI.R256.02 – Updated wording. |
| | | adelong2: LBI.R106.03 – Added “up to” to programming mode time. |
| | | adelong2: LBI.R317.02 – Updated to new TP RspStatus. |
| | | adelong2: LBI.R318.02 – Updated to new TP RspStatus. Changed HMI behavior. |
| | | adelong2: LBI.R109.03 – Updated wording. |
| | | adelong2: LBI.R259.02 – Updated to new TP RspStatus. |
| | | adelong2: LBI.R260.02 – Added “backup” to the requirement. |
| | | adelong2: LBI.R357.01 – New. Addresses valet password deletion failure. |
| | | adelong2: LBI.R358.01 – New. Addresses valet password deletion failure. |
| | | adelong2: LBI.R269.03 – Corrected error regarding source. Added note. |
| | | adelong2: LBI.R270.02 – Updated wording. |
| | | adelong2: LBI.R271.02 – Added “up to” to programming mode time. |
| | | adelong2: LBI.R320.02 – Updated to new TP RspStatus. |
| | | adelong2: LBI.R321.02 – Updated to new TP RspStatus. Changed HMI behavior |
| | | adelong2: LBI.R276.03 – Updated wording. |
| | | adelong2: Updated use case for deleting valet password. Removed “with password” from pre-condition about PaaK devices in vehicle. |
| | | adelong2: Updated sequence diagrams for deleting valet passwords. Added missing CAN signals. Updated title to remove “with backup password”. |
| | | adelong2: LBI.R277.02 – Added “hashes” to passwords. |
| | | adelong2: LBI.R278.02 – Updated wording. Added “hashes” to passwords. Included logic for determining when to send delete all requests to BCM based on keypad status. |
| | | adelong2: LBI.R279.02 – Added “up to” to programming mode time. |
| | | adelong2: LBI.R282.02 – Updated wording. Added “hash” to password. Included logic for determining when to send delete requests to BCM based on keypad status. |
| | | adelong2: LBI.R359.01 – LBI.R362.01 – New requirements for this function, but essentially duplicates from deleting the backup password function. |
| | | adelong2: LBI.R129.04 – Removed erroneous conditions (i.e. non-Crypto search ones). |
| | | adelong2: LBI.R323.02 – Updated wording. |
| | | adelong2: LBI.R363.01 – LBI.R371.01– New requirements for this function, but essentially duplicates from starting the vehicle function. |
| | | adelong2: LBI.R324.02 – Corrected error regarding search type. |
| | | adelong2: LBI.R325.02 – Changed HMI behavior. |
| | | adelong2: LBI.R283.02 – Corrected error regarding search type. |
| | | adelong2: LBI.R286.02 – Added “backup” to the requirement. |
| | | adelong2: LBI.R287.02 – Added requirement for BLEM to keep track of whether a device/key index has an associated keypad code, which is used for determining when to send requests to BCM. |
| | | adelong2: LBI.R289.02 – Updated wording. |
| | | adelong2: LBI.R372.01 – New. Addresses behavior when swapping BLEM in vehicle. |

| | | |
|---|---|---|
| | | adelong2: LBI.R293.02 – Clarified that this requirement pertains only to requests to store the valet keypad code. Added “up to” to programming mode time. |
| | | adelong2: LBI.R294.03 – Clarified that this requirement pertains only to requests to store the valet keypad code. |
| | | adelong2: Changed APIM to SYNC in section 3.1.3 for consistency. |
| | | adelong2: LBI.R131.03 – Updated wording. |
| | | adelong2: LBI.R132.03 – Updated wording. Removed lockout condition. |
| | | adelong2: LBI.R373.01 – New. Addresses HMI behavior when functions are in process and conditions change. |
| | | adelong2: LBI.R374.01 – New. Defines conditions for activating Enhanced Valet mode. |
| | | adelong2: LBI.R375.01 – New. Addresses HMI behavior when function is in process and conditions change. |
| | | adelong2: LBI.R133.02 – Updated wording. |
| | | adelong2: LBI.R134.04 – Updated wording. |
| | | adelong2: LBI.R135.03 – Updated wording. |
| | | adelong2: LBI.R376.01 – New. Defines conditions for deactivating Enhanced Valet mode. |
| | | adelong2: LBI.R298.02 – Removed because redundant. |
| | | adelong2: LBI.R299.02 – Removed because no longer applies. |
| | | adelong2: LBI.R136.02 – Updated wording. |
| | | adelong2: LBI.R377.01 – New. Addresses HMI behavior when function is in process and conditions change. |
| | | adelong2: LBI.R179.03 – Removed “with passwords”. PaaK devices are not required to have an associated backup password to be used for authenticating Enhanced Valet Mode. |
| | | adelong2: LBI.R378.01 – New. Replaces duplicate of R180. |
| | | adelong2: LBI.R176.04 – Added item to notification payload. Corrected error. |
| | | adelong2: LBI.R177.03 – Removed timestamp from BLEM message. TCU will timestamp the message. Updated note. |
| | | adelong2: Corrected CAK acronym. |
| | | adelong2: Added reference documents and revision numbers. |
| Nov 29, 2017 | 2.5 | Fifth Release |
| | | adelong2: LBI.R006.02 – Changed note to specify that the name of a PaaK device is the device name generated during PaaK setup. |
| | | adelong2: LBI.R325.03 – Added a line that specifies how long the message will display in the HMI and when it should be dismissed. |
| | | adelong2: LBI.R328.02 – Deleted. Delay no longer necessary. |
| | | ekarpins: LBI.R379.02 – Modified. Defines how SYNC should structure the keypad code when it transmits it to the BLEM. |
| Feb 6, 2018 | 2.6 | ekarpins: LBI.R380.01 – New requirement to define feature behavior after unsuccessful Master/PaaK reset. To be implemented at a later time. |
| | | ekarpins: LBI.R381.01 – New requirement explaining CCS(Customer Connectivity Settings) impact to LBI feature functionality. |
| | | ekarpins: LBI.R382.01 – New requirement explaining BLEM notifications to SDN when a successful Master or PaaK Reset takes place. |
| | | ekarpins: LBI.R322.02 – Modify requirement to include engine status instead of ignition status as input signal. May need to add additional info wrt Privacy Mode/Vehicle Connectivity. |

| | | |
|---|---|---|
| | | ekarpins: LBI.R325.04 – Modify requirement to include engine status instead of ignition status as input signal when determining which screen to display to customer when engine is running in NonMotive mode (vehicle is in secure idle or remote started state). |
| | | ekarpins: LBI.R383.01 – New requirement to include engine status instead of ignition status as input signal when determining which screen to display to customer when engine is not running. |
| June 8, 2018 | 2.7 | ekarpins: LBI.R384.01-R394.01 – Added section 3.1.4 to describe interface requirements for Interior Registry Search (LBI Key Fob Search). |
| | | ekarpins: LBI.R395.01-R403.01 – Added section 3.1.5 to describe BLEM-BCM interface requirements for Keypad Programming. |
| | | Ekarpins: LBI.R146.03 – added requirements for minimum acceptable password length for FNV2 architecture/FB5 (feature bundle 5, SYNC 4) |
| | | Ekarpins: LBI.R381.02 – added requirement for when privacy mode is enabled/Connectivity disabled by customer and how it affects the feature. |
| | | Ekarpins: LBI.R404.01 – added requirements explaining SYNC behavior when Privacy Mode is ON/Connectivity OFF |
| | | ~~Ekarpins:LBI.R405.01 – added requirements explaining BLEM behavior when Privacy Mode is ON.~~ |
| | | Ekarpins: LBI.R379.02 – modified requirements for # of keypad digits based on the market and how keypad presses bit-mapped. |
| | | Ekarpins: LBI.R288.03 – modified requirements for # of keypad digits and added additional info. |
| | | Ekarpins: LBI.R407.01 – added requirements explaining BLEM behavior while in lockout mode – ignores any opcode requests. |
| | | Ekarpins: LBI.R408.01 – added requirements to allow BLEM verify preconditions like ignition status and transmission status are matched with SYNC |
| | | Ekarpins: LBI.R292.02 – modified BCM requirement to verify vehicle config when storing keypad code. |
| | | Ekarpins: LBI.R083.03 – modified requirements related to valet and number of digits |
| | | Ekarpins: LBI.R097.04 – modified req to include details about number of digits/keypad and valet mode |
| | | Ekarpins: LBI.R237.03 – modified req related to number of keypad digits |
| | | Ekarpins: LBI.R238.03 – modified req related to number of keypad digits |
| | | Ekarpins: LBI.R081.04 – modified req related to number of keypad digits |
| | | Ekarpins: LBI.R103.05 – modified reqs related to privacy Mode/Vehicle Connectivity |
| | | ekarpins: LBI.R322.03 – Added additional requirements wrt Vehicle connectivity/Privacy Mode |
| | | Ekarpins: LBI.R289.03 - Modified reqs wrt 8/10 digits valet passcode |
| | | Ekarpins: LBI.R189.03, 200.03, 308.02, 206.03, 231.03, 252.04, 241.02, 248.03, 315.02, 255.02, 269.03 – modified by adding additional notes to explain Key ID and BLEM SyncPPacket |
| | | Ekarpins: LBI.R151.02, 158.02, 159.02 – modified req to include clarification for SHA256 and calculation example. |
| | | Ekarpins: LBI.R409.01 – new req for SYNC to verify keypad code restrictions if the vehicle is for European market |
| | | Ekarpins: LBI.R178.02 – modified req to clarify backup password control rules/ranking |

| | | |
|---|---|---|
| October 18, 2018 | 2.8 | Ekarpins: **LBI.R093.03 – updated req with additional info related to creation of EVM with Backup password** |
| | | Ekarpins: **LBI.R410.01 – new req to localize Paak phones with 2.5s de-bounce timer for when creating backup passwords** |
| Last req is R410 | | |
| | | |

# 1 Architectural Design

## 1.1 Feature Overview

PaaK Backup serves as a backup method for starting vehicles that are equipped with Phone-as-a-Key (PaaK). This feature utilizes the existing keypad entry system as well as a new, password-based starting system. It allows the customer to start and drive away their vehicle even if their phone is not functional (e.g. drained battery) or if their phone is lost, stolen or destroyed. If any of these situations occur, customers can gain entry to the vehicle via the keypad and then use the SYNC HMI to enter a password to prime the vehicle for starting. PaaK Backup also allows customers to generate a temporary password and keypad code to give to valet attendants through an Enhanced Valet Mode in SYNC. This functionality will be integrated into the Valet Mode feature of SYNC.

## 1.2 Assumptions and Constraints

PaaK Backup will be offered as an opt-in feature. Users may set it up at any time, but they must first activate a phone-as-a-key. PaaK allows users to activate up to four mobile phones as keys. PaaK Backup will allow the customer to associate a unique backup password and keypad code with each of these four phones-as-keys. The BLEM will associate all backup passwords to PaaK key indexes/CAKs authorized for the given vehicle. The valet password will be given a separate key index. The BCM will associate all personalized keypad codes with PaaK key indexes.

Creating and using passwords with PaaK Backup does not require the vehicle to have cloud connectivity. However, cloud connectivity is needed to enhance the security of PaaK Backup. Having connectivity will allow the vehicle to send notifications to the user whenever a password is added, deleted, or used at the vehicle. For PaaK, cloud connectivity to the vehicle and the phone is required to allow the user to revoke their PaaK key remotely. This key revoke action will remove the key from the vehicle as well as remove the backup password and keypad code associated with that key.

The PaaK Backup system is distributed over multiple vehicle subsystems and has functions that are performed by more than one ECU on the vehicle. The BLEM, which is the main controller of the PaaK system, will be responsible for password storage and verification. The BCM will store personal keypad codes and challenge the BLEM when the user attempts to start the vehicle. SYNC will function as the interface for creating and using the passwords. The SYNC module itself will not manage these passwords but rather serve as a pass-through to the BLEM.

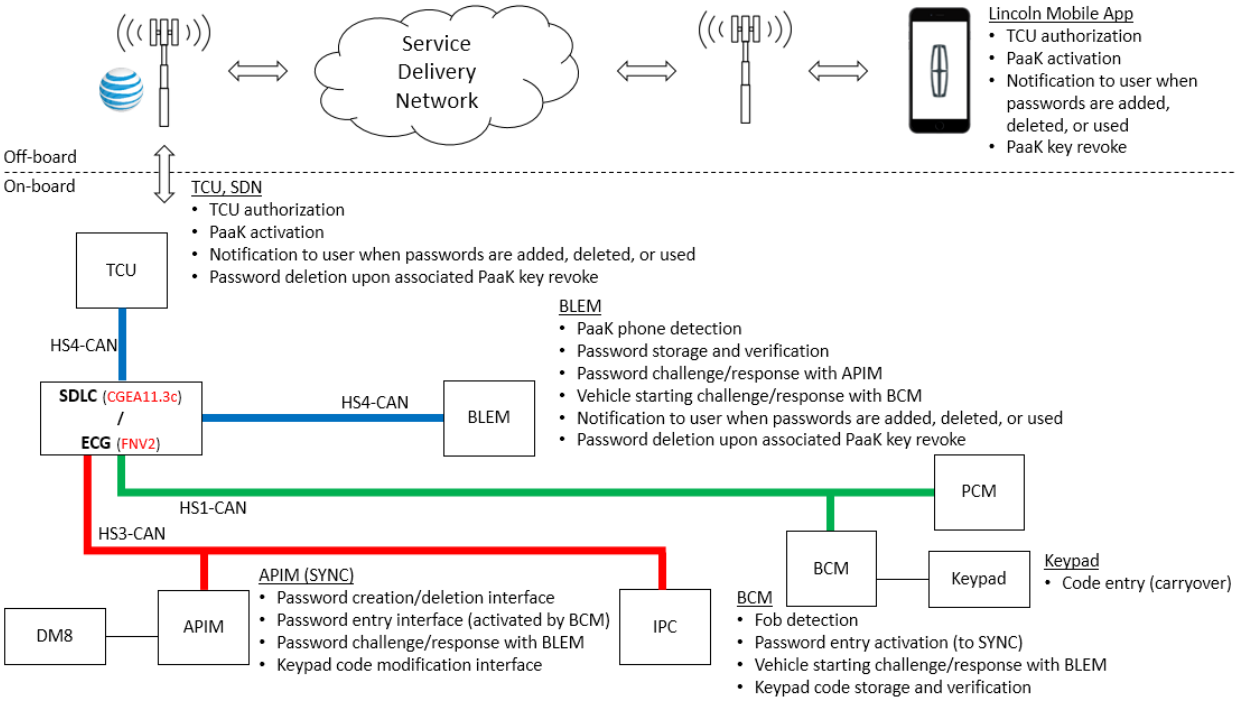
Note: This is a list of required CAN signals. See TP APIM/BLEM SPSS for a list of required TP methods.

| Sig. ID | Signal Name | Signal Description | Encoding | | Min | Max | Tx | Rx |
|---|---|---|---|---|---|---|---|---|
| 1 | IgnPsswrdActv_B_Stat | Password activity status | 0x0 | Inactive | 0 (0x0) | 1 (0x1) | BLEM | BCM, APIM |
| | | | 0x1 | Active | | | | |
| 2 | IgnPsswrdDsply_B_Rq (wakeup signal) | Password entry screen trigger | 0x0 | Inactive | 0 (0x0) | 1 (0x1) | BCM | APIM |
| | | | 0x1 | Active | | | | |
| 3 | FobTrgtPssvData_No_Rq | 5 bytes of data (challenge data) | Unitless | | 0 (0x0) | 1099511627775 (0xFFFFFFFFFF) | BLEM | BCM |
| 4 | FobCtlPssvData_No_Actl | 5 bytes of data (response data) | Unitless | | 0 (0x0) | 1099511627775 (0xFFFFFFFFFF) | BCM | BLEM |
| 5 | FobTrgtType_D_Rq | Type of search being requested | 0x0 | Null | 0 (0x0) | 7 (0x7) | BLEM | BCM |
| | | | 0x1 | Crypto | | | | |
| | | | 0x2 | Registry | | | | |
| | | | 0x3 | Polling | | | | |
| | | | 0x4 - 0x7 | *NotUsed* | | | | |
| 6 | FobCtlType_D_Stat | Indication of whether a valid key fob was found during the last search | 0x0 | Null | 0 (0x0) | 3 (0x3) | BCM | BLEM |
| | | | 0x1 | Invalid | | | | |
| | | | 0x2 | Valid | | | | |
| | | | 0x3 | *NotUsed* | | | | |
| 7 | FobTrgtZone_D_Rq | Indication of the zone in or around the vehicle to be searched | 0x0 | Null | 0 (0x0) | 15 (0xF) | BLEM | BCM |
| | | | 0x1 | Interior | | | | |
| | | | 0x2 | Driver | | | | |
| | | | 0x3 | Passenger | | | | |
| | | | 0x4 | RearExterior | | | | |
| | | | 0x5 | RearInterior | | | | |
| | | | 0x6 | Approach | | | | |
| | | | 0x7 - 0xF | *Reserved* | | | | |
| 8 | FobTrgtRollCode_No_Rq | Rolling count transmitted by the Target function to align a search request with the corresponding search result | | Unitless | 0 (0x0) | 15 (0xF) | BLEM | BCM |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9 | FobCtlRollCode_No_Stat | Rolling count transmitted by the Control function to align a search request with the corresponding search result | | Unitless | 0 (0x0) | 15 (0xF) | BCM | BLEM |
| 10 | IgnPsswrdSetup_B_Rq | Signal used to trigger LBI setup | 0x0 | Inactive | 0 (0x0) | 1 (0x1) | BLEM | APIM |
| | | | 0x1 | Active | | | | |
| 11 | IgnPsswrdLckout_B_Stat | Signal used to enable lockout | 0x0 | Inactive | 0 (0x0) | 1 (0x1) | BLEM | APIM |
| | | | 0x1 | Active | | | | |
| 12 | KeyPadCodeProg_D_Rq | Signal used to trigger KeyPad Code Add/Delete request | 0x0 | Null | 0 (0x0) | 4 (0x4) | BLEM | BCM |
| | | | 0x1 | ProgrammingMode | | | | |
| | | | 0x2 | Add | | | | |
| | | | 0x3 | Delete | | | | |
| | | | 0x4 | DeleteAll | | | | |
| 13 | KeyPadCodeProg_D_Stat | Signal indicating status for Keypad Code Add/Delete request | 0x0 | NormalMode | 0 (0x0) | 5 (0x5) | BCM | BLEM |
| | | | 0x1 | LearningMode | | | | |
| | | | 0x2 | Add | | | | |
| | | | 0x3 | Delete | | | | |
| | | | 0x4 | DeleteAll | | | | |
| | | | 0x5 | ProgrammingFailure | | | | |
| | | | 0x6 | Duplicate | | | | |
| 14 | PaakTrgtActvData_No_Rq | RKE Challenge Data | | Unitless | 0 (0x0) | 1099511627775 (0xFFFFFFFFFF) | BCM | BLEM |
| 15 | PaakCtlActvData_No_Actl | RKE Response Data | | Unitless | 0 (0x0) | 1099511627775 (0xFFFFFFFFFF) | BLEM | BCM |
| 16 | PaaKCtlActv_No_Actl | Phone SubID | | Unitless | 0 (0x0) | 63 (0x3F) | BLEM | BCM |
| 17 | PaakTrgtType_D_Rq | Type of search being requested | 0x0 | Null | 0 (0x0) | 7 (0x7) | BCM | BLEM |
| | | | 0x1 | Crypto | | | | |
| | | | 0x2 | Registry | | | | |
| | | | 0x3 | Polling | | | | |
| 18 | GearLvrPos_D_Actl | Gear lever State | 0x0 | Park | 0 (0x0) | 63 (0x3F) | TCM | BLEM/ APIM |
| | | | 0x1 | Reverse | | | | |
| | | | 0x2 | Neutral | | | | |
| | | | 0x3 | Drive | | | | |
| | | | 0x4 | Sport_DriveSport | | | | |
| | | | 0x5 | Low | | | | |
| | | | 0x6 | First | | | | |
| | | | 0x7 | Second | | | | |
| | | | 0x8 | Third | | | | |
| | | | 0x9 | Fourth | | | | |
| | | | 0xA | Fifth | | | | |
| | | | 0xB | Sixth | | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 0xC | Undefined_Treat_as_Fault | | | | |
| | | | 0xD | Undefined_Treat_as_Fault | | | | |
| | | | 0xE | Unknown_Position | | | | |
| | | | 0xF | Fault | | | | |
| 19 | Ignition_Status | Status of Ignition | 0x0 | Unknown | 0 (0x0) | 15 (0xF) | BCM | BLEM/ APIM |
| | | | 0x1 | Off | | | | |
| | | | 0x2 | Accessory | | | | |
| | | | 0x4 | Run | | | | |
| | | | 0x8 | Start | | | | |
| | | | 0xF | Invalid | | | | |
| 20 | FactoryReset_Rq | Factory Reset Defaults | 0x0 | Inactive | 0 (0x0) | 1 (0x1) | APIM | BLEM |
| | | | 0x1 | ResetFactoryDefaults | | | | |
| 21 | Delay_Accy | Delayed Accessory Mode | 0x0 | Off | 0 (0x0) | 1 (0x1) | BCM | APIM |
| | | | 0x1 | On | | | | |
| 22 | ModemReset_D_Rq | Modem reset | 0x0 | Null | 0 (0x0) | 15 (0xF) | APIM | BLEM |
| | | | 0x1 | WifiHotSpot_Reset | | | | |
| | | | 0x2 | Paak_Reset | | | | |
| | | | 0x3 | Online_Traffic_Reset | | | | |
| | | | 0x4 | CCS_Reset | | | | |
| | | | 0x5 - 0xF | NotUsed | | | | |
| 23 | PwPckTq_D_Stat | Motive/Non-motive vehicle status | 0x00 | PwPckOff_TqNotAvailable | 0 (0x0) | 3 (0x3) | PCM | APIM |
| | | | 0x01 | PwPckOn_TqNotAvailable | | | | |
| | | | 0x02 | StartInPrgrss_TqNotAvailable | | | | |
| | | | 0x03 | PwPckOn_TqAvailable | | | | |

## 1.3 System Diagram




# 2 Functional Definition

Note: The actual Marketing names for PaaK and PaaK Backup are yet to be determined. References to PaaK and PaaK Backup are for descriptive purposes.

## 2.1 Primary Functions

### 2.1.1 Creating backup password and keypad code for PaaK device

#### *2.1.1.1 Requirements*

**[LBI.R001.03]** The BLEM shall keep track of whether PaaK devices in the vehicle have an associated backup password. If the BLEM detects one such device while ignition is in Run (*Ignition_Status = 0x4 = Run*) and transmission is in Park (*GearLvlPos_D_Actl = 0x0 = Park*), it shall notify SYNC (*IgnPsswrdSetup_B_Rq = 0x1 = Active*) *five times,* once per ignition cycle, per lifetime of the CAK associated with that device.

**[LBI.R002.02]** When SYNC receives *IgnPsswrdSetup_B_Rq = 0x1 = Active* from the BLEM, the SYNC HMI shall display a message to the user with the option to create a backup password.

**[LBI.R003.03]** When the user selects the option to create a backup password, the SYNC HMI shall display a screen with requirements and steps to create a backup password.

**[LBI.R330.01]** When user chooses to continue, then SYNC shall query the BLEM for PaaK devices without passwords in the vehicle and shall request the cryptographic salt (*BackupIgnition_Rq* with *OpCode = 0x04 = Salt and Check for PaaK without Passwords, Byte 5 = 0x00*).

**[LBI.R183.02]** The BLEM shall trigger a BCM Interior Registry search (*FobTrgtType_D_Rq = 0x2 = Registry, FobTrgtZone_D_Rq = 0x1 = Interior, FobTrgtPssvData_No_Rq, FobTrgtRollCode_No_Rq*) whenever it receives *BackupIgnition_Rq* with *OpCode = 0x04 = Salt and Check for PaaK without Passwords.*

***Note:*** *The encrypted data is used to authenticate searches. The roll code is used to synchronize the request and response.*
***See details of BCM Interior Registry search in Section 3.1.4***

**[LBI.R184.02]** After completing BLEM-requested Interior Registry search (*FobTrgtType_D_Rq = 0x2 = Registry, FobTrgtZone_D_Rq = 0x1 = Interior, FobTrgtPssvData_No_Rq, FobTrgtRollCode_No_Rq*), the BCM shall report to the BLEM whether any key fobs were detected in the vehicle (*FobCtlType_D_Stat, FobCtlPssvData_No_Actl, FobCtlRollCode_No_Stat*).

**[LBI.R006.02]** After receiving Interior Registry search results from the BCM, the BLEM shall report to SYNC what devices (PaaK w/o passwords or key fob) were found in the vehicle, the names and key indexes of all PaaK devices found in the vehicle, as well as the cryptographic salt (*BackupIgnition_Rsp* with *RspCode = 0x04 = Salt and Check for PaaK without Passwords Response*).

***Note:*** *Name here refers to the device name generated during PaaK setup.*

**[LBI.R007.02]** When SYNC receives *BackupIgnition_Rsp* with *RspCode = 0x04 = Salt and Check for PaaK without Passwords Response*, the SYNC HMI shall display either:

1. Error message that lists key fob as missing with options to retry or cancel if:
    - *RspStatus = 0x02 = One PaaK w/o Password and No Fob In Vehicle* OR
    - *RspStatus = 0x05 = Two+ PaaK w/o Password and No Fob In Vehicle* AND
    - *Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*
2. Error message that lists phone as missing with options to retry or cancel if:
    - *RspStatus = 0x03 = Fob In Vehicle and No PaaK* AND
    - *Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*
3. Error message that lists key fob and phone as missing with options to retry or cancel if:
    - *RspStatus = 0x06 = No PaaK w/o Password and No Fob In Vehicle* AND
    - *Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*

4. Backup password creation screen if:
    - *RspStatus = 0x01 = One PaaK w/o Password and Fob In Vehicle* AND
    - *Byte 6 = 0x01, Challenge Nonce = EOS, Salt = Salt, Valet Password = EOS, KeyIndex = KeyIndex, PhoneName = PhoneName*
5. List of all detected PaaK devices with instruction for the user to choose the desired device if:
    - *RspStatus = 0x04 = Two+ PaaK w/o Password and Fob In Vehicle* AND
    - *Byte 6 = 0x02 - 0x04, Challenge Nonce = EOS, Salt = Salt, Valet Password = EOS, KeyIndex = KeyIndex, PhoneName = PhoneName*

**[LBI.R008.02]** When the user chooses their desired device from the list, the SYNC HMI shall display the backup password creation screen.

**[LBI.R009.02]** The SYNC HMI shall require the user to enter their new backup password twice. If passwords do not match, the SYNC HMI shall notify the user and provide an option to retry.

**[LBI.R010.02]** SYNC shall check entered passwords against password requirements in real time. The SYNC HMI shall not allow the user to proceed to the next screen until their password meets the minimum requirements.

**[LBI.R011.03]** SYNC shall compute a hash of the entered password and send the result to the BLEM with the key index of the selected device (*BackupIgnition_Rq* with *OpCode = 0x07 = Password Transmit, KeyIndex = KeyIndex, Password = Password, KeypadCode = EOS*).

**[LBI.R012.03]** Upon receiving the password hash and key index (*BackupIgnition_Rq* with *OpCode = 0x07 = Password Transmit*), the BLEM shall trigger a BCM Interior Registry search (*FobTrgtType_D_Rq = 0x2 = Registry, FobTrgtZone_D_Rq = 0x1 = Interior, FobTrgtPssvData_No_Rq, FobTrgtRollCode_No_Rq*).

**[LBI.R185.02]** After completing BLEM-requested Interior Registry search (*FobTrgtType_D_Rq = 0x2 = Registry, FobTrgtZone_D_Rq = 0x1 = Interior, FobTrgtPssvData_No_Rq, FobTrgtRollCode_No_Rq*), the BCM shall report to the BLEM whether any key fobs were detected in the vehicle (*FobCtlType_D_Stat, FobCtlPssvData_No_Actl, FobCtlRollCode_No_Stat*).

**[LBI.R013.02]** After receiving Interior Registry search results from the BCM, the BLEM shall respond to SYNC with *BackupIgnition_Rsp* with *RspCode = 0x07 = Password Response* and either
1. *RspStatus = 0x08 = Fob No Longer Detected* if:
    - The BCM does not detect a key fob in the vehicle (*FobCtlType_D_Stat = 0x1 = Invalid*) AND
    - The BLEM detects the PaaK device associated with the received key index.
2. *RspStatus = 0x07 = PaaK No Longer Detected* if:
    - The BCM detects a key fob in the vehicle (*FobCtlType_D_Stat = 0x2 = Valid*) AND
    - The BLEM does not detect the PaaK device associated with the received key index
3. *RspStatus = 0x09 = PaaK and Fob No Longer Detected* if:

- The BCM does not detect a key fob in the vehicle (*FobCtlType_D_Stat = 0x1 = Invalid*) AND
- The BLEM does not detect the PaaK device associated with the received key index

***Note: See details of BCM Interior Registry search in Section 3.1.4***

**[LBI.R186.01]** When SYNC receives *BackupIgnition_Rsp* with *RspCode = 0x07 = Password Response*, the SYNC HMI shall display either:

1. Error message that lists key fob as missing with options to retry (restart the process) or cancel. if:
   - *RspStatus = 0x08 = Fob No Longer Detected* AND
   - *Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*
2. Error message that lists phone as missing with options to retry (restart the process) or cancel. if:
   - *RspStatus = 0x07 = PaaK No Longer Detected* AND
   - *Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*
3. Error message that lists key fob and phone as missing with options to retry (restart the process) or cancel if:
   - *RspStatus = 0x09 = PaaK and Fob No Longer Detected* AND
   - *Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*

**[LBI.R014.02]** If the PaaK device that the user selected and a key fob are still in the vehicle when the BLEM receives the password hash, the BLEM shall verify that the entered password is not already being used.

**[LBI.R187.01]** If the BLEM determines that the password is already being used, the BLEM shall notify SYNC of this (*BackupIgnition_Rsp* with *RspCode = 0x07 = Password Response, RspStatus = 0x0A = Password Already Used, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R188.01]** When SYNC receives *BackupIgnition_Rsp* with *RspCode = 0x07 = Password Response, RspStatus = 0x0A = Password Already Used,* the SYNC HMI shall display a message that password is already being used and instruct user to enter a different password.

**[LBI.R015.02]** If the BLEM determines that password is not already being used, the BLEM shall store the password hash in its HSM and associate it with the received key index.

**[LBI.R331.01]** When the BLEM cannot store the password hash in its HSM, it shall notify SYNC (*BackupIgnition_Rsp with RspCode = 0x07 = Password Response, RspStatus = 0x0C = Password Created Failed, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R332.01]** When SYNC receives creation failure response (*BackupIgnition_Rsp* with *RspCode = 0x07 = Password Response, RspStatus = 0x0C = Password Created Failed*), the SYNC HMI shall notify user of unsuccessful password creation.

**[LBI.R189.03]** When the BLEM stores a new password hash, it shall report this to the TCU (*LBIAlert_St* with *Event = 0x01 = Backup Password Created, Source = 0x00, [timestamp], [Key ID]*) in a BLEM SyncP signed packet (Service Type 0x40/Sub-Service 0x0) as defined in Transfer Protocol BLEM SPSS.

***Note:*** *Reference TP BLEM SPSS and LBI SPSS for BLEM SyncPPacket definition and its payload. Key ID here refers to the Key ID of the PaaK device associated with the created backup password.*

**[LBI.R190.01]** When the BLEM stores a new password hash, it shall report this to SYNC (*BackupIgnition_Rsp* with *RspCode = 0x07 = Password Response, RspStatus = 0x0B = Password Created Successfully, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R191.01]** When SYNC receives *BackupIgnition_Rsp* with *RspCode = 0x07 = Password Response, RspStatus = 0x0B = Password Created Successfully*, the SYNC HMI shall notify the user of successful backup password creation.

**[LBI.R016.01]** After creating backup password, the SYNC HMI shall present the user with the option to set up a personal keypad code.

**[LBI.R017.02]** If the user chooses not to create a personal keypad code, the SYNC HMI shall inform the user that PaaK Backup setup is complete and provide instructions on how to use the feature.

**[LBI.R018.02]** If the user chooses to create a personal keypad code, the SYNC HMI shall display a screen for entering a new personal keypad code.

**[LBI.R019.01]** The SYNC HMI shall require the user to enter their new personal keypad code twice.

**[LBI.R192.01]** SYNC shall verify that the two codes entered match. If the codes do not match, the SYNC HMI shall notify the user and provide an option to retry.

**[LBI.R020.02]** If the codes match, SYNC shall send to the BLEM the keypad code, the key index of the selected PaaK device, and a request to store the keypad code (*BackupIgnition_Rq* with *OpCode = 0x08 = Keypad Code Create Request, KeyIndex = KeyIndex, Password = EOS, KeypadCode = KeypadCode*).

**[LBI.R021.02]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x08 = Keypad Code Create Request, KeyIndex = KeyIndex, Password = EOS, KeypadCode = KeypadCode*, it shall respond (*PaaKCtlActvData_No_Actl*) to periodic RKE challenge from the BCM (*PaaKTrgtActvData_No_Rq*) and send a request to the BCM to enter keypad programming mode (*KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode*).

***Note: See details of BLEM-BCM Keypad programming requirements in Section 3.1.5***

**[LBI.R022.03]** When the BCM receives *KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode* together with valid RKE response data from the BLEM, it shall enter keypad programming mode and notify the BLEM (*KeyPadCodeProg_D_Stat = 0x2 = LearningMode*) within two seconds and remain in programming mode/provide notification for up to two seconds.

**[LBI.R023.02]** When the BLEM receives *KeyPadCodeProg_D_Stat = 0x2 = LearningMode*, it shall send to the BCM the received keypad code (in *PaaKCtlActvData_No_Actl*), the received key index (*PaaKCtlActv_No_Actl = [index]*) and a request to store the personal keypad code (*KeyPadCodeProg_D_Rq = 0x2 = Add*).

**[LBI.R024.02]** When the BCM receives the request to store the personal keypad code (*KeyPadCodeProg_D_Rq = 0x2 = Add*) together with the key index (*PaaKCtlActv_No_Actl = [index]*) and the keypad code (in *PaaKCtlActvData_No_Actl*), it shall store the received keypad code and associate it with the received key index.

**[LBI.R193.01]** When the BCM stores a new keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x3 = Add*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

**[LBI.R302.01]** When the BCM cannot store a new keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x5 = ProgrammingFailture*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

**[LBI.R025.03]** When the BLEM receives confirmation of keypad code storage (*KeyPadCodeProg_D_Stat = 0x3 = Add*), it shall:

- Update keypad code association status for selected/detected key index
- Notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x08 = Keypad Code Create Response, RspStatus = 0x0D = Keypad Code Created Successfully, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R194.01]** When SYNC receives confirmation of keypad code storage (*BackupIgnition_Rsp* with *RspCode = 0x08 = Keypad Code Create Response, RspStatus = 0x0D = Keypad Code Created Successfully*), the SYNC HMI shall notify the user of successful keypad code creation.

**[LBI.R333.01]** After SYNC notifies user of successful keypad code creation, it shall inform user that PaaK Backup setup is complete and provide instructions on how to use the feature.

**[LBI.R303.01]** When the BLEM receives programming failure response (*KeyPadCodeProg_D_Stat = 0x3 = 0x5 = ProgrammingFailure*), it shall notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x08 = Keypad Code Create Response, RspStatus = 0x0E = Keypad Code Created Failed, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R304.01]** When SYNC receives programming failure response (*BackupIgnition_Rsp* with *RspCode = 0x08 = Keypad Code Create Response, RspStatus = 0x0E = Keypad Code Created Failed*), the SYNC HMI shall notify user of programming failure.

#### *2.1.1.2 Use Case*

##### 2.1.1.2.1 Creating backup password and keypad code for PaaK device

| | |
|---|---|
| **Actors** | User |
| **Pre-conditions** | User has previously activated Phone-as-a-Key feature for vehicle.<br>Vehicle is in RUN.<br>User is inside vehicle.<br>One associated PaaK device and key fob are inside the vehicle. |
| **Scenario Description** | 1. User selects option to Create Backup Password for Phone-as-a-Key from PaaK Backup Settings in SYNC.<br>2. SYNC displays screen with creation steps.<br>3. SYNC displays alphanumeric password entry screen and instructs user to enter a backup password.<br>4. User enters password twice according to password requirements.<br>5. User selects Enter.<br>6. SYNC displays message that backup password has been created successfully. SYNC also asks user if they would like to create a personal keypad code.<br>7. User selects option to create a personal keypad code.<br>8. SYNC displays screen for entering personal keypad code.<br>9. User enters personal keypad code twice.<br>10. SYNC displays message that new personal keypad code has been created successfully. |

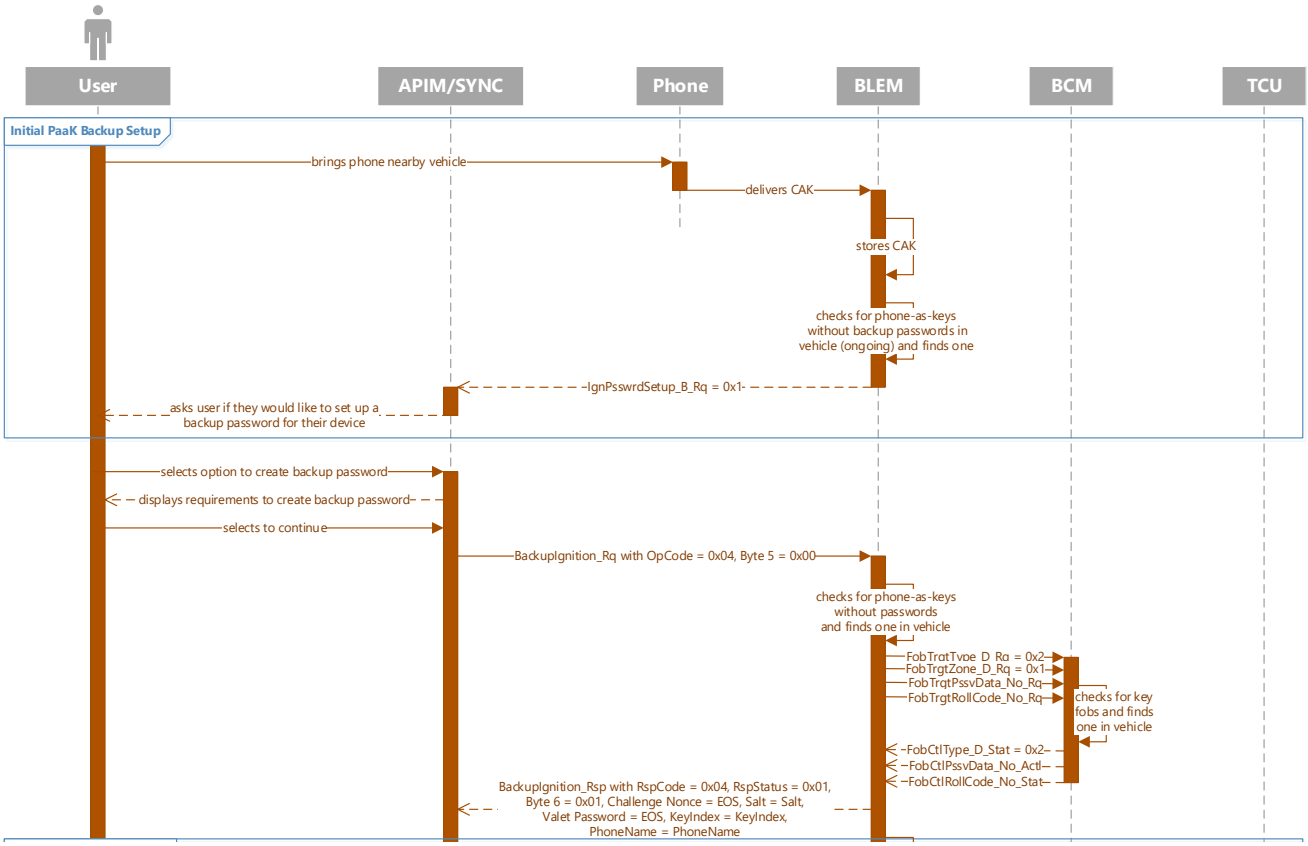
| | |
|---|---|
| **Post-conditions** | PaaK Backup is ready for use.<br>Notification that backup password has been created is sent to user. |
| **List of Exceptions** | User enters password that does not meet requirements.<br>User enters passwords that do not match.<br>User enters passwords that is already in use. |
| **Interfaces** | APIM<br>BCM<br>BLEM<br>TCU<br>SDN<br>PaaK FI |

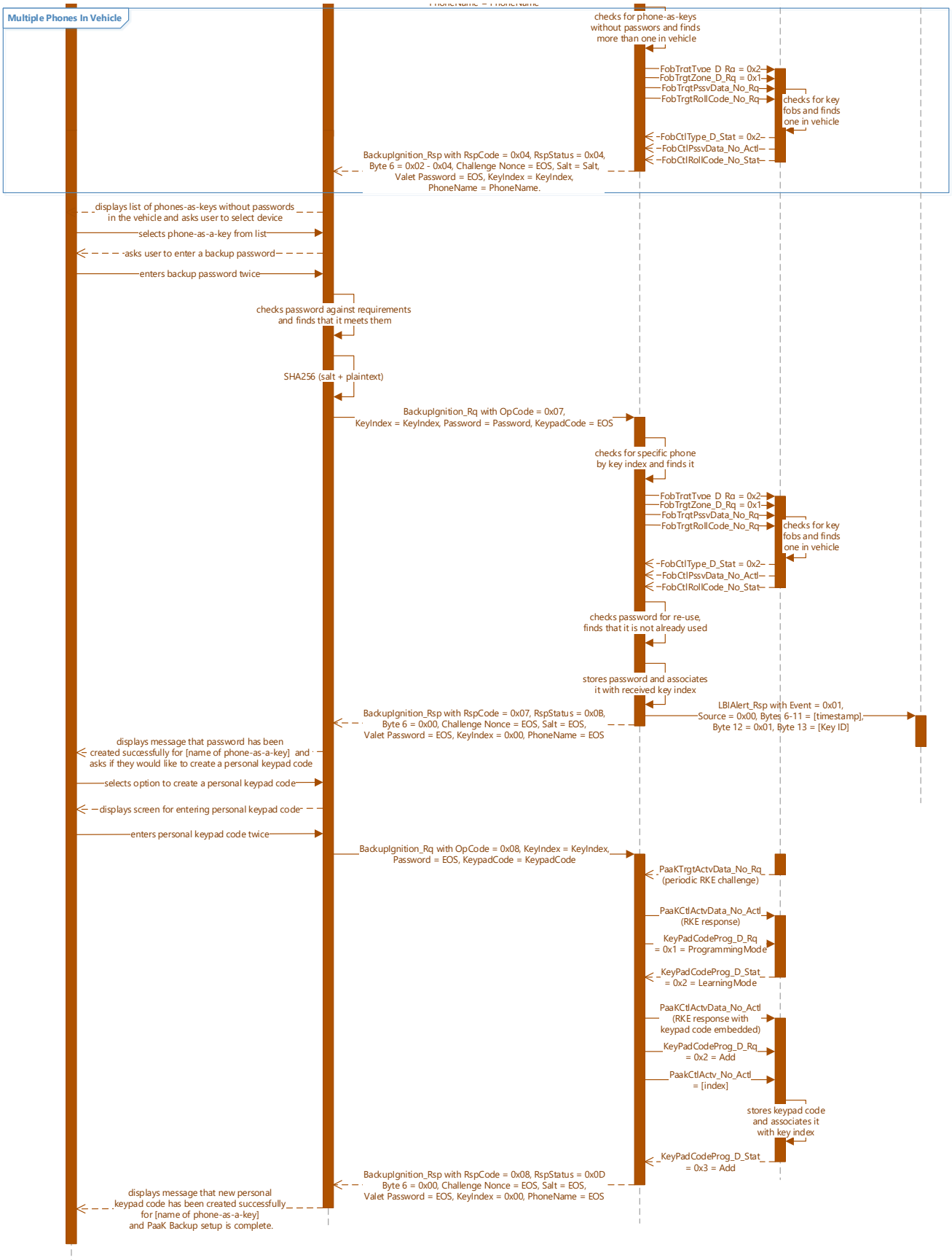
### *2.1.1.3 Sequence Diagrams*

#### 2.1.1.3.1 Creating backup password and keypad code for PaaK device – happy path

Pre-conditions:

1. User has previously activated a PaaK device for their vehicle.
2. Vehicle is in RUN and user is inside vehicle.
3. BLEM has vehicle-unique salt.

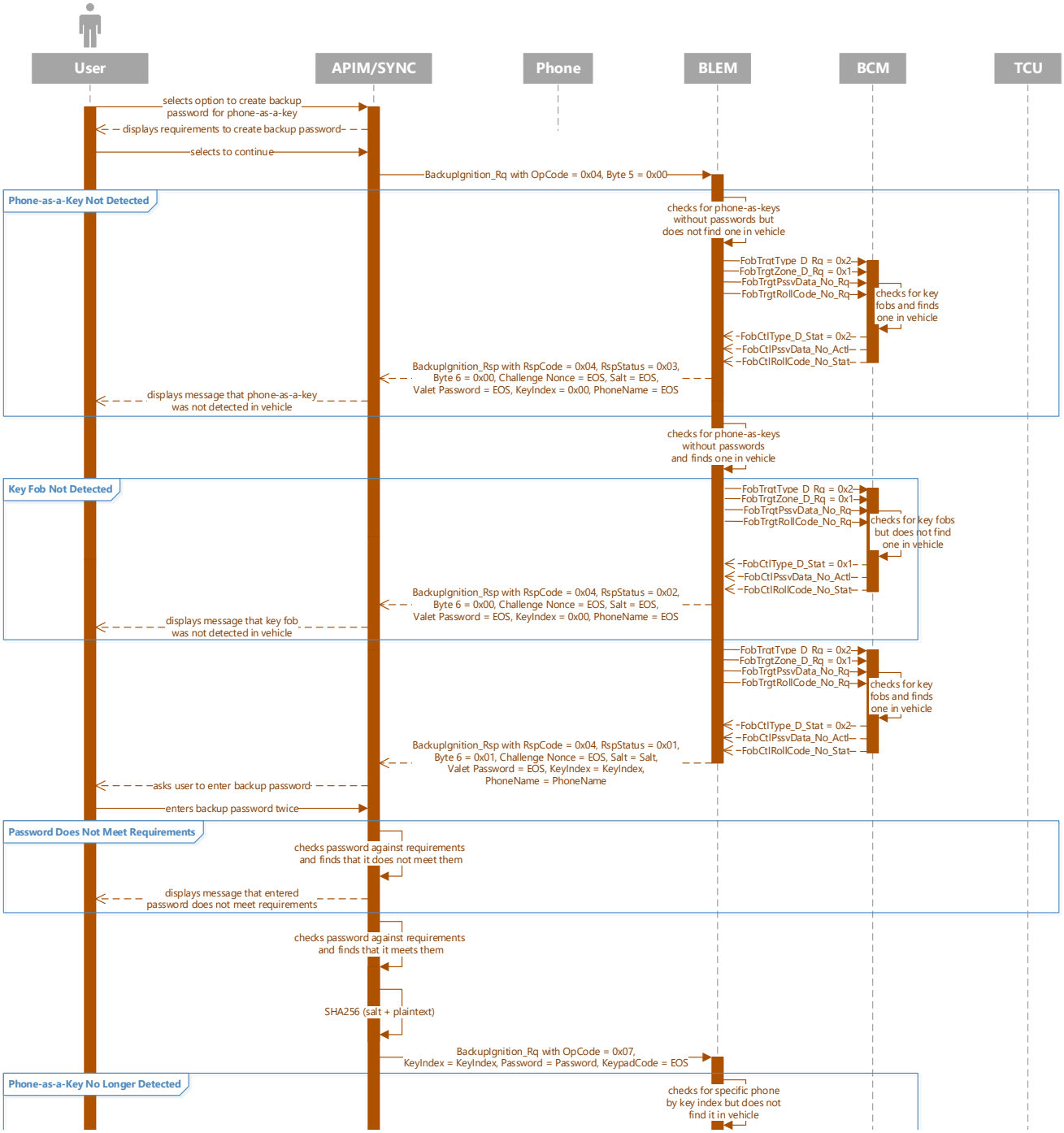


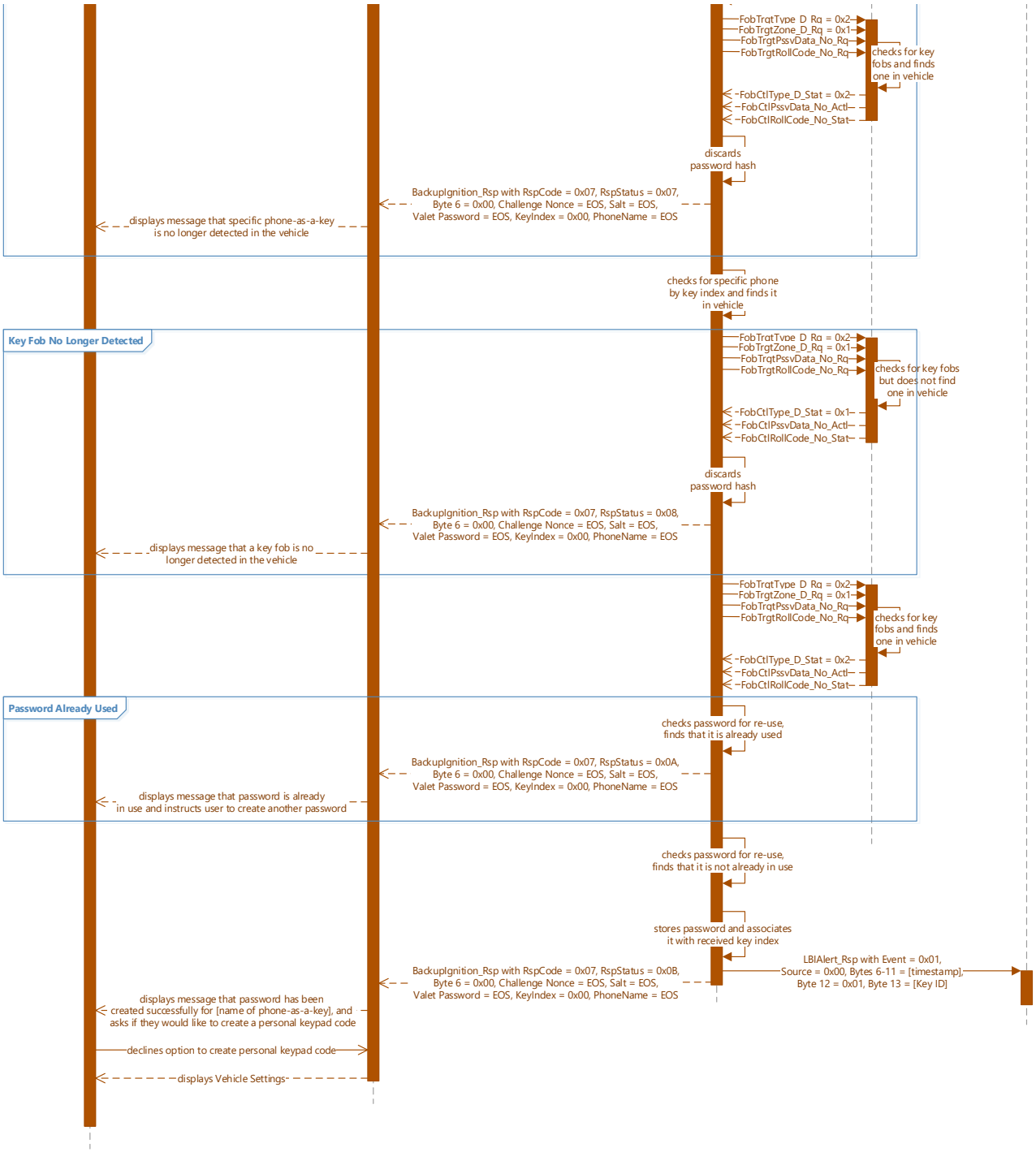

Multiple Phones In Vehicle
checks for phone-as-keys without passwords and finds more than one in vehicle
FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs and finds one in vehicle
FobCtlType_D_Stat = 0x2
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
BackupIgnition_Rsp with RspCode = 0x04, RspStatus = 0x04, Byte 6 = 0x02 - 0x04, Challenge Nonce = EOS, Salt = Salt, Valet Password = EOS, KeyIndex = KeyIndex, PhoneName = PhoneName.
displays list of phones-as-keys without passwords in the vehicle and asks user to select device
selects phone-as-a-key from list
asks user to enter a backup password
enters backup password twice
checks password against requirements and finds that it meets them
SHA256 (salt + plaintext)
BackupIgnition_Rq with OpCode = 0x07, KeyIndex = KeyIndex, Password = Password, KeypadCode = EOS
checks for specific phone by key index and finds it
FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs and finds one in vehicle
FobCtlType_D_Stat = 0x2
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
checks password for re-use, finds that it is not already used
stores password and associates it with received key index
BackupIgnition_Rsp with RspCode = 0x07, RspStatus = 0x0B, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
LBIAlert_Rsp with Event = 0x01, Source = 0x00, Bytes 6-11 = [timestamp], Byte 12 = 0x01, Byte 13 = [Key ID]
displays message that password has been created successfully for [name of phone-as-a-key] and asks if they would like to create a personal keypad code
selects option to create a personal keypad code
displays screen for entering personal keypad code
enters personal keypad code twice
BackupIgnition_Rq with OpCode = 0x08, KeyIndex = KeyIndex, Password = EOS, KeypadCode = KeypadCode
PaaKTrgtActvData_No_Rq (periodic RKE challenge)
PaaKCtlActvData_No_Actl (RKE response)
KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode
KeyPadCodeProg_D_Stat = 0x2 = LearningMode
PaaKCtlActvData_No_Actl (RKE response with keypad code embedded)
KeyPadCodeProg_D_Rq = 0x2 = Add
PaaKCtlActv_No_Actl = [index]
stores keypad code and associates it with key index
KeyPadCodeProg_D_Stat = 0x3 = Add
BackupIgnition_Rsp with RspCode = 0x08, RspStatus = 0x0D Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
displays message that new personal keypad code has been created successfully for [name of phone-as-a-key] and PaaK Backup setup is complete.

#### 2.1.1.3.2 Creating backup password and keypad code for PaaK device – error scenarios

Pre-conditions:

1. User has previously activated a PaaK device for their vehicle.
2. Vehicle is in RUN and user is inside vehicle.
3. BLEM has vehicle-unique salt.




FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs and finds one in vehicle
FobCtlType_D_Stat = 0x2
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
discards password hash
BackupIgnition_Rsp with RspCode = 0x07, RspStatus = 0x07, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
displays message that specific phone-as-a-key is no longer detected in the vehicle
checks for specific phone by key index and finds it in vehicle
Key Fob No Longer Detected
FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs but does not find one in vehicle
FobCtlType_D_Stat = 0x1
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
discards password hash
BackupIgnition_Rsp with RspCode = 0x07, RspStatus = 0x08, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
displays message that a key fob is no longer detected in the vehicle
FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs and finds one in vehicle
FobCtlType_D_Stat = 0x2
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
Password Already Used
checks password for re-use, finds that it is already used
BackupIgnition_Rsp with RspCode = 0x07, RspStatus = 0x0A, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
displays message that password is already in use and instructs user to create another password
checks password for re-use, finds that it is not already in use
stores password and associates it with received key index
LBIAlert_Rsp with Event = 0x01, Source = 0x00, Bytes 6-11 = [timestamp], Byte 12 = 0x01, Byte 13 = [Key ID]
BackupIgnition_Rsp with RspCode = 0x07, RspStatus = 0x0B, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
displays message that password has been created successfully for [name of phone-as-a-key], and asks if they would like to create a personal keypad code
declines option to create personal keypad code
displays Vehicle Settings

### 2.1.2 Starting vehicle with backup password

#### *2.1.2.1 Requirements*

**[LBI.R026.02]** The BCM shall notify SYNC (*IgnPsswrdDsply_B_Rq = 0x1 = Active*) whenever:

1. The user presses the start button or the brake pedal AND
2. No key fobs or phones-as-keys are detected in the vehicle AND
3. There is at least one backup password created (*IgnPsswrdActv_B_Stat = 0x1 = Active*).

**[LBI.R027.04]** When SYNC receives *IgnPsswrdDsply_B_Rq = 0x1 = Active* and *Ignition_Status = 0x1 = Off* and the status of Enhanced Valet Mode in SYNC is inactive, SYNC shall enter Infotainment Mode and display either:

1. A backup password entry screen if *IgnPsswrdLckout_B_Stat = 0x0 = Inactive* OR
2. A lockout popup if *IgnPsswrdLckout_B_Stat = 0x1 = Active*

**[LBI.R197.02]** When the user enters a password at the backup password entry screen, SYNC shall request a challenge from the BLEM (*BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request, Byte 5 = 0x00*).

**[LBI.R198.01]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request*, it shall issue a challenge to SYNC with cryptographic nonce and salt (*BackupIgnition_Rsp* with *RspCode = 0x01 = Issue Challenge, RspStatus = 0x00 = Reserved, Byte 6 = 0x00, Challenge Nonce = Challenge Nonce, Salt = Salt, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R199.01]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request*, it shall compute, using the cryptographic nonce, another hash of all stored password hashes.

**[LBI.R031.02]** SYNC shall compute a hash of entered password using received salt and then compute a hash of this result using received nonce.

**[LBI.R032.02]** SYNC shall respond to the challenge from the BLEM (*BackupIgnition_Rsp* with *RspCode = 0x01 = Issue Challenge*) with computed password hash (*BackupIgnition_Rq* with *OpCode = 0x02 = Challenge Response, KeyIndex = EOS, Password = Challenge Password, KeypadCode = EOS*).

**[LBI.R033.01]** When the BLEM receives a challenge hash from SYNC (*BackupIgnition_Rq* with *OpCode = 0x02 = Challenge Response, KeyIndex = EOS, Password = Challenge Password, KeypadCode = EOS*), it shall compare it with the hashes that it computed for the stored passwords.

**[LBI.R034.03]** If the BLEM determines that the received password is valid i.e. challenge hash matches a calculated password hash, it shall notify SYNC of this (*BackupIgnition_Rsp with RspCode = 0x02 = Challenge Response Acknowledge, RspStatus = 0x0F = Valid Password, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*) and start a 21-second authorization period.

**[LBI.R307.02]** When the BLEM is in a 21-second authorization period while *Ignition_Status = 0x1 = Off*, the BLEM shall respond positively (*PaakCtlType_D_Stat = 0x1 = Valid, PaakCtlIndx1_No_Actl = [Index]*) to BCM Crypto Start searches (*PaakTrgtType_D_Rq = 0x1 = Crypto, PaakTrgtZone_D_Rq = 0x1 = Interior)*, but negatively to Registry or Polling searches. After this period expires, the BLEM shall respond negatively (*PaakCtlType_D_Stat = 0x1 = Invalid*) to BCM Crypto Start searches.

*Note: The index that the BLEM sends is the key index of the PaaK device that the entered password is associated with.*

**[LBI.R200.03]** When the BLEM responds positively to a Crypto Start search during 21-second authorization period and it is not in Enhanced Valet Mode, it shall report this to the TCU (*LBIAlert_St* with *Event = 0x02 = Backup Password Used, Source = 0x00, [timestamp], [Key ID]*) in a BLEM SyncP signed packet (Service Type 0x40/Sub-Service 0x0) as defined in Transfer Protocol BLEM SPSS.

***Note:*** *Reference TP BLEM SPSS and LBI SPSS for BLEM SyncPPacket definition and its payload. Key ID here refers to the Key ID of the PaaK device associated with the entered backup password.*

**[LBI.R035.02]** When SYNC receives a valid notification from the BLEM (*BackupIgnition_Rsp with RspCode = 0x02 = Challenge Response Acknowledge, RspStatus = 0x0F = Valid Password*), the SYNC HMI shall notify the user that the entered password has been accepted and that they must start the vehicle now. This message shall display for 20 seconds unless vehicle ignition status changes to Run (*Ignition_Status = 0x4 = Run*), then this message shall be dismissed.

**[LBI.R036.02]** If the BLEM determines that the received password is invalid i.e. challenge hash does not match a calculated password hash, it shall increment invalid password counter and then notify SYNC (*BackupIgnition_Rsp with RspCode = 0x02 = Challenge Response Acknowledge, RspStatus = 0x10 = Invalid Password, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R037.03]** When SYNC receives an invalid password notification (*BackupIgnition_Rsp with RspCode = 0x02 = Challenge Response Acknowledge, RspStatus = 0x10 = Invalid Password*), the SYNC HMI shall notify the user that the entered password is invalid and provide an option to retry.

**[LBI.R038.01]** The BLEM shall keep track of invalid attempts at entering the backup password and invalid attempts at entering the valet password in separate counters.

**[LBI.R201.03]** Invalid attempt counters shall be stored in NVM and shall not be reset by change in vehicle ignition status, network status, or battery state of charge.

**[LBI.R039.01]** The backup password counter shall be reset each time a backup password is successfully entered. The valet password counter shall be reset each time a valet password is successfully entered or Enhanced Valet Mode is exited.

**[LBI.R040.03]** If either attempt counter reaches five invalid attempts, the BLEM shall start a five minute timer and notify SYNC of lockout (*BackupIgnition_Rsp with RspCode = 0x02 = Challenge Response Acknowledge, RspStatus = 0x19 = Lockout, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*) and send out *IgnPsswrdLckout_B_Stat = Active* until timer expires.

**[LBI.R407.01]** When the BLEM initiates a five minute lockout timer it shall ignore any incoming opcode requests until lockout timer expires.

**[LBI.R308.02]** When the BLEM starts a lockout timer, it shall report this to the TCU (*LBIAlert_St* with *Event = 0x07 = Lockout, Source = 0x00, [timestamp], [Key ID]*) in a BLEM SyncP signed packet (Service Type 0x40/Sub-Service 0x1) as defined in Transfer Protocol BLEM SPSS.

***Note:*** *Reference TP BLEM SPSS and LBI SPSS for BLEM SyncPPacket definition and its payload. Key ID here should be left blank, zero. This is a "notify all" alert since it is impossible to determine which Key ID is being used if invalid passwords are continuously entered. No Key ID's can be contained since it is not possible to know which one is the primary target, so this alert needs to be sent out to users of all PaaK devices.*

**[LBI.R202.03]** Lockout timer shall be stored in NVM and shall not be reset by change in vehicle ignition status, network status, or battery state of charge.

**[LBI.R334.01]** During these five minutes, whenever the user attempts to use a password to start the vehicle, exit secure idle, reset a current password, or activate/deactivate Enhanced Valet Mode, the SYNC HMI shall display a screen with information that password entry has been locked out.

**[LBI.R041.02]** Lockout shall occur every time the attempt counter reaches five invalid attempts.

#### *2.1.2.2 Use Case*

##### 2.1.2.2.1 Starting vehicle with backup password

| **Actors** | User |
|---|---|
| **Pre-conditions** | User has previously created a backup password.<br>Vehicle is locked.<br>User is outside vehicle.<br>No associated key fobs or phones-as-keys are near the vehicle. |

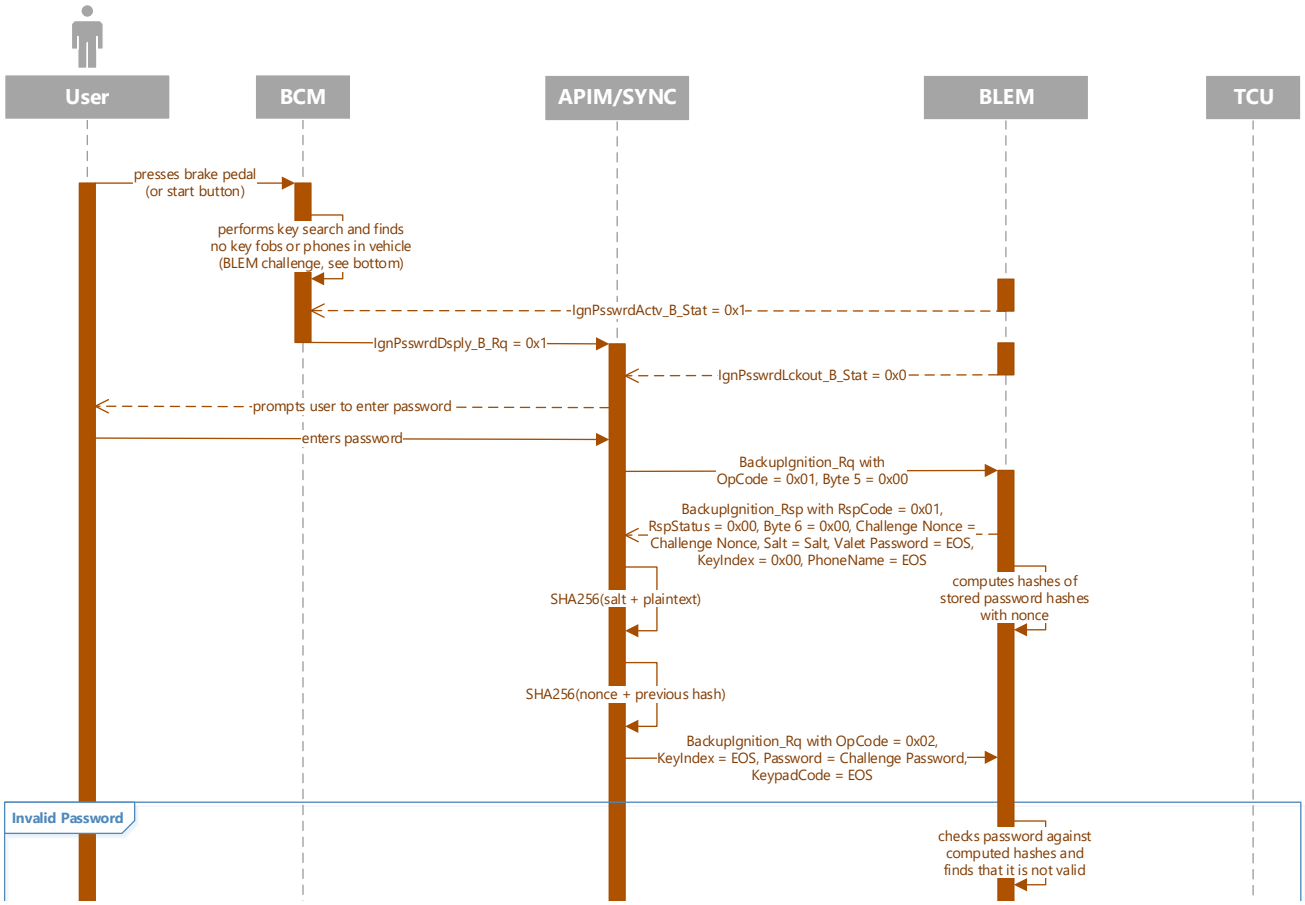
| | |
|---|---|
| **Scenario Description** | 1. User approaches vehicle.<br>2. User enters valid keypad code.<br>3. Vehicle unlocks.<br>4. User opens door and enters vehicle.<br>5. User presses brake pedal.<br>6. SYNC displays backup password entry screen.<br>7. Without being inactive for more than 30 seconds, user enters valid backup password via SYNC. (This includes inputting the password then selecting Enter.)<br>8. SYNC displays message instructing user to start the vehicle.<br>9. Within 20 seconds, user presses start button while holding brake pedal.<br>10. Vehicle starts with engine running. |
| **Post-conditions** | User is able to drive away vehicle.<br>User is able to charge their PaaK device in vehicle.<br>Notification that PaaK Backup has been used is sent to user. |
| **List of Exceptions** | User does not enter valid keypad code.<br>User does not enter valid password.<br>User is inactive for more than 30 seconds while SYNC displays password entry screen.<br>User does not start vehicle within 20 seconds of successful password entry.<br>User presses start button without holding brake pedal after password is accepted. |
| **Interfaces** | APIM<br>BCM<br>BLEM<br>TCU<br>SDN<br>PaaK FI |

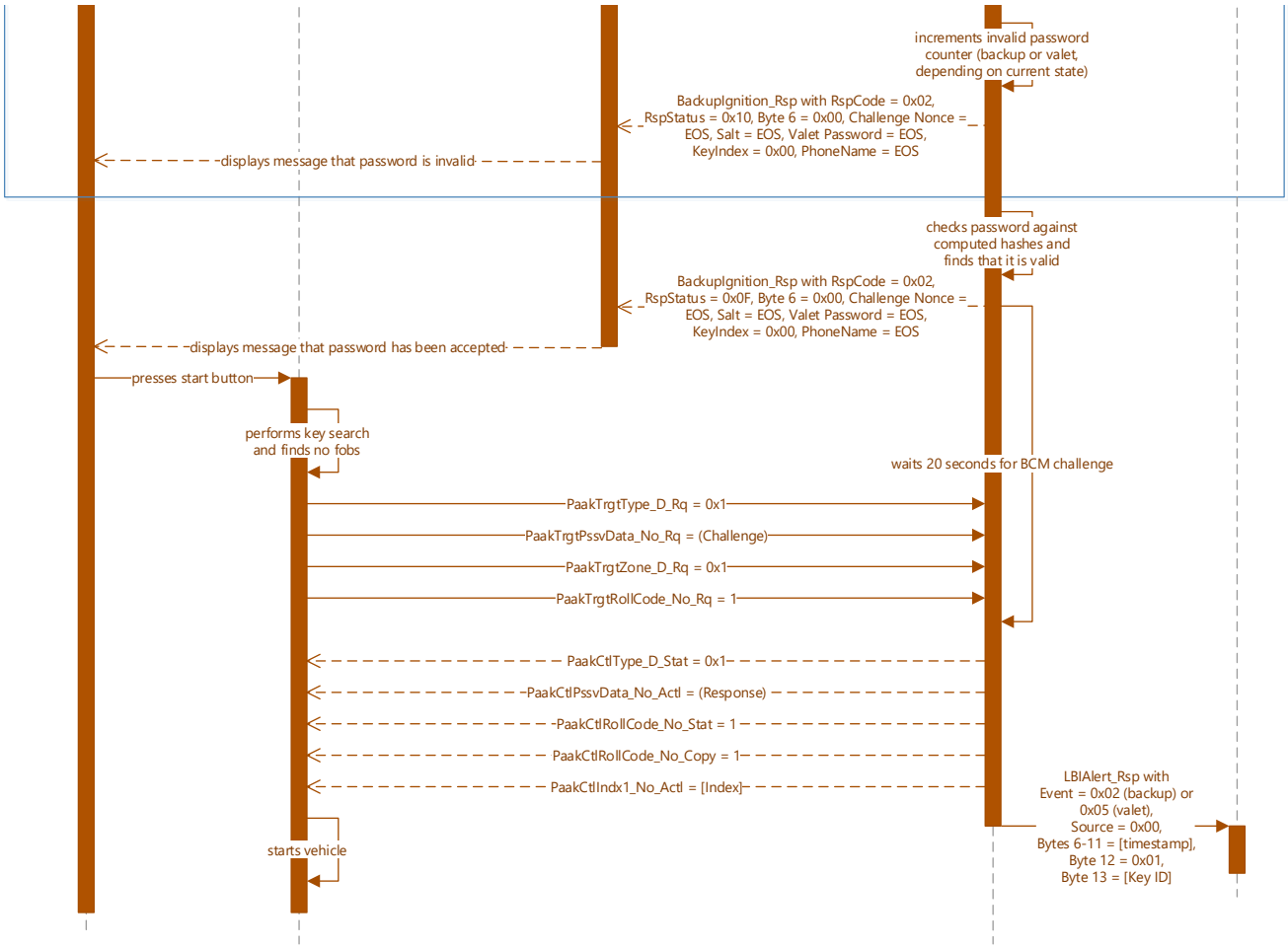
#### *2.1.2.3 Sequence Diagram*

##### 2.1.2.3.1 Starting vehicle with backup password

Pre-conditions

1. User has previously created backup password.
2. Vehicle is off, unlocked.
3. No associated key fobs or PaaK devices are present in or around vehicle.
4. BLEM has vehicle-unique salt.


User
BCM
APIM/SYNC
BLEM
TCU
presses brake pedal
(or start button)
performs key search and finds
no key fobs or phones in vehicle
(BLEM challenge, see bottom)
IgnPsswrdActv_B_Stat = 0x1
IgnPsswrdDsply_B_Rq = 0x1
IgnPsswrdLckout_B_Stat = 0x0
prompts user to enter password
enters password
BackupIgnition_Rq with
OpCode = 0x01, Byte 5 = 0x00
BackupIgnition_Rsp with RspCode = 0x01,
RspStatus = 0x00, Byte 6 = 0x00, Challenge Nonce =
Challenge Nonce, Salt = Salt, Valet Password = EOS,
KeyIndex = 0x00, PhoneName = EOS
computes hashes of
stored password hashes
with nonce
SHA256(salt + plaintext)
SHA256(nonce + previous hash)
BackupIgnition_Rq with OpCode = 0x02,
KeyIndex = EOS, Password = Challenge Password,
KeypadCode = EOS
Invalid Password
checks password against
computed hashes and
finds that it is not valid


increments invalid password counter (backup or valet, depending on current state)
BackupIgnition_Rsp with RspCode = 0x02, RspStatus = 0x10, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
displays message that password is invalid
checks password against computed hashes and finds that it is valid
BackupIgnition_Rsp with RspCode = 0x02, RspStatus = 0x0F, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
displays message that password has been accepted
presses start button
performs key search and finds no fobs
waits 20 seconds for BCM challenge
PaakTrgtType_D_Rq = 0x1
PaakTrgtPssvData_No_Rq = (Challenge)
PaakTrgtZone_D_Rq = 0x1
PaakTrgtRollCode_No_Rq = 1
PaakCtlType_D_Stat = 0x1
PaakCtlPssvData_No_Actl = (Response)
PaakCtlRollCode_No_Stat = 1
PaakCtlRollCode_No_Copy = 1
PaakCtlIndx1_No_Actl = [Index]
LBIAlert_Rsp with Event = 0x02 (backup) or 0x05 (valet), Source = 0x00, Bytes 6-11 = [timestamp], Byte 12 = 0x01, Byte 13 = [Key ID]
starts vehicle

### 2.1.3 Deleting backup password and keypad code for PaaK device

#### *2.1.3.1 Requirements*

**[LBI.R042.02]** When the user selects the option to delete a backup password within SYNC settings, SYNC shall query the BLEM for PaaK devices with passwords in the vehicle and shall request the cryptographic salt (*BackupIgnition_Rq* with *OpCode = 0x03 = Salt and Check for PaaK with Passwords, Byte 5 = 0x00*).

**[LBI.R203.02]** The BLEM shall trigger a BCM Interior Registry search (*FobTrgtType_D_Rq = 0x2 = Registry, FobTrgtZone_D_Rq = 0x1 = Interior, FobTrgtPssvData_No_Rq, FobTrgtRollCode_No_Rq*) whenever it receives *BackupIgnition_Rq* with *OpCode = 0x03 = Salt and Check for PaaK with Passwords*.

***Note: See details of BCM Interior Registry search in Section 3.1.4***

**[LBI.R204.02]** After completing BLEM-requested Interior Registry search (*FobTrgtType_D_Rq = 0x2 = Registry, FobTrgtZone_D_Rq = 0x1 = Interior, FobTrgtPssvData_No_Rq, FobTrgtRollCode_No_Rq*), the BCM shall report to the BLEM whether any key fobs were detected in the vehicle (*FobCtlType_D_Stat, FobCtlPssvData_No_Actl, FobCtlRollCode_No_Stat*).

**[LBI.R205.02]** After receiving Interior Registry search results from the BCM, the BLEM shall report to SYNC what devices (PaaK with passwords or key fob) were found in the vehicle, the names and key indexes of all PaaK devices found in the vehicle, as well as the cryptographic salt (*BackupIgnition_Rsp* with *RspCode = 0x03 = Salt and Check for PaaK with Passwords Response*).

**[LBI.R043.02]** When SYNC receives *BackupIgnition_Rsp* with *RspCode = 0x03 = Salt and Check for PaaK with Passwords Response*, the SYNC HMI shall display either:

1. Error message that lists phone as missing with options to retry or cancel if:
    - *RspStatus = 0x13 = Fob In Vehicle and No PaaK w/ Password* OR
    - *RspStatus = 0x16 = No PaaK w/ Password and No Fob In Vehicle* AND
    - *Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*
2. List of all detected PaaK devices with instruction for the user to choose the desired device if:
    - *RspStatus = 0x11 = One PaaK w/ Password and Fob In Vehicle* OR
    - *RspStatus = 0x12 = One PaaK w/ Password and No Fob In Vehicle* OR
    - *RspStatus = 0x14 = Two+ PaaK w/ Password and Fob In Vehicle* OR
    - *RspStatus = 0x15 = Two+ PaaK w/ Password and No Fob In Vehicle* AND
    - *Byte 6 = 0x01 – 0x04, Challenge Nonce = EOS, Salt = Salt, Valet Password = EOS, KeyIndex = KeyIndex, PhoneName = PhoneName*

**[LBI.R046.03]** When the user chooses their desired device (password) for deletion, the SYNC HMI shall ask the user to confirm deletion of the associated password.

**[LBI.R047.03]** When the user confirms deletion of their password, SYNC shall send to the BLEM the key index of the selected PaaK device together with a request to delete the password hash associated with this key index (*BackupIgnition_Rq* with *OpCode = 0x09 = Password Delete Request, KeyIndex = KeyIndex, Password = EOS, KeypadCode = EOS*).

**[LBI.R048.03]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x09 = Password Delete Request, KeyIndex = KeyIndex, Password = EOS, KeypadCode = EOS*, it shall immediately delete the password hash associated with the received key index.

**[LBI.R206.03]** When the BLEM deletes a backup password hash, it shall report this to the TCU (*LBIAlert_St* with *Event = 0x03 = Backup Password Deleted, Source = 0x00, [timestamp], [Key ID]*) in a BLEM SyncP signed packet (Service Type 0x40/Sub-Service 0x0) as defined in Transfer Protocol BLEM SPSS.

***Note:*** *Reference TP BLEM SPSS and LBI SPSS for BLEM SyncPPacket definition and its payload. Key ID here refers to the Key ID of the PaaK device associated with the deleted backup password.*

**[LBI.R207.02]** When the BLEM deletes a backup password hash, it shall also determine whether there is a keypad code associated with the received key index.

***Note: See details of BLEM-BCM Keypad programming requirements in Section 3.1.5***

If there is an associated keypad code, the BLEM shall initiate keypad code deletion by responding (with *PaaKCtlActvData_No_Actl*) to periodic RKE challenge from the BCM (*PaaKTrgtActvData_No_Rq*) and sending a request to the BCM to enter keypad programming mode (*KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode*).

If there is no associated keypad code, the BLEM shall notify SYNC of successful deletion (*BackupIgnition_Rsp* with *RspCode = 0x09 = Password Delete Response, RspStatus = 0x17 = Password Deleted Successfully, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R335.01]** When the BLEM cannot delete the backup password hash, it shall notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x09 = Password Delete Response, RspStatus = 0x18 = Password Deleted Failed, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R336.01]** When SYNC receives deletion failure response (*BackupIgnition_Rsp* with *RspCode = 0x09 = Password Delete Response, RspStatus = 0x18 = Password Deleted Failed*), the SYNC HMI shall notify user of unsuccessful password deletion.

**[LBI.R049.03]** When the BCM receives *KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode* together with valid RKE response data from the BLEM, it shall enter keypad programming mode and notify the BLEM (*KeyPadCodeProg_D_Stat = 0x2 = LearningMode*) within two seconds and continue to provide notification for up to two seconds.

**[LBI.R050.02]** When the BLEM receives *KeyPadCodeProg_D_Stat = 0x2 = LearningMode*, it shall send to the BCM the received key index (*PaaKCtlActv_No_Actl = [index]*) and a request to delete the personal keypad code (*KeyPadCodeProg_D_Rq = 0x3 = Delete*).

**[LBI.R051.02]** When the BCM receives the request to delete the personal keypad code (*KeyPadCodeProg_D_Rq = 0x3 = Delete*) together with the key index (*PaaKCtlActv_No_Actl = [index]*), it shall delete the personal keypad code that is associated with the received key index.

**[LBI.R208.01]** When the BCM deletes a keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x4 = Delete*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

**[LBI.R309.01]** When the BCM cannot delete a new keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x5 = ProgrammingFailture*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

**[LBI.R052.02]** When the BLEM receives confirmation of keypad code deletion (*KeyPadCodeProg_D_Stat = 0x4 = Delete*), it shall notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x09 = Password Delete Response, RspStatus = 0x17 = Password Deleted Successfully, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R209.01]** When SYNC receives confirmation of keypad code deletion (*BackupIgnition_Rsp* with *RspCode = 0x09 = Password Delete Response, RspStatus = 0x17 = Password Deleted Successfully*), the SYNC HMI shall notify the user of successful password (and keypad code, if applicable) deletion.

**[LBI.R310.02]** When the BLEM receives programming failure response (*KeyPadCodeProg_D_Stat = 0x5 = ProgrammingFailure*), it shall notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x09 = Password Delete Response, RspStatus = 0x1E = Password Deleted Successfully, but Keypad Code Deleted Failed, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R311.02]** When SYNC receives programming failure response (*BackupIgnition_Rsp* with *RspCode = 0x09 = Password Delete Response, RspStatus = 0x1E = Password Deleted Successfully, but Keypad Code Deleted Failed*), the SYNC HMI shall notify user of successful password deletion and unsuccessful keypad code deletion.

#### *2.1.3.2 Use Case*

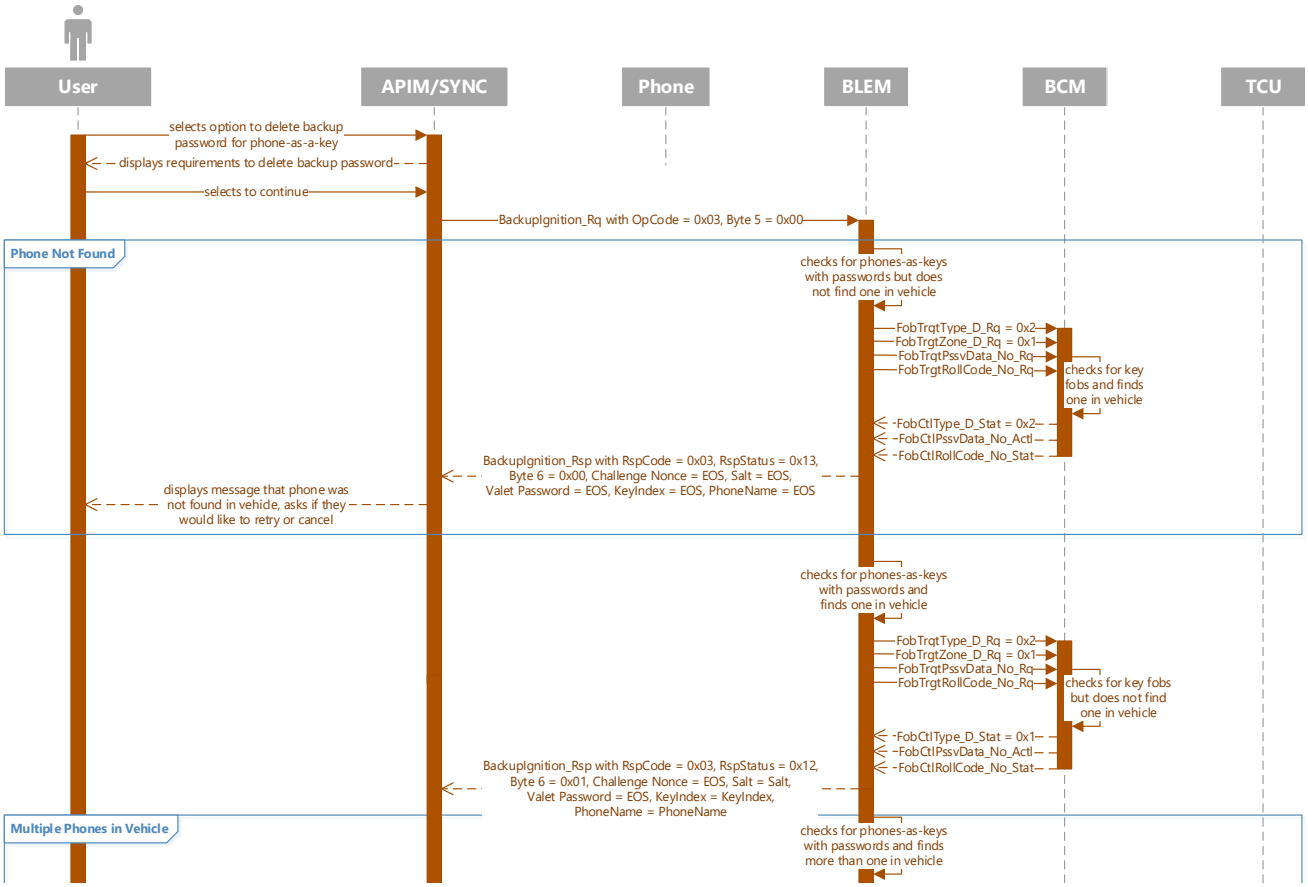
| | |
|---|---|
| **Actors** | User |
| **Pre-conditions** | Vehicle is in RUN.<br>User is inside vehicle.<br>One associated PaaK device with password is inside the vehicle. |
| **Scenario Description** | 1. User selects option to Delete Backup Password for Phone-as-a-Key from PaaK Backup Settings in SYNC.<br>2. SYNC displays message with deletion requirements.<br>3. User continues.<br>4. SYNC displays message asking user if they are sure they want to delete their password and personal keypad code.<br>5. User confirms.<br>6. SYNC displays message that backup password and personal keypad code have been deleted successfully. |
| **Post-conditions** | Notification that backup password has been deleted is sent to user. |
| **List of Exceptions** | |
| **Interfaces** | APIM<br>BCM<br>BLEM<br>TCU<br>SDN<br>PaaK FI |

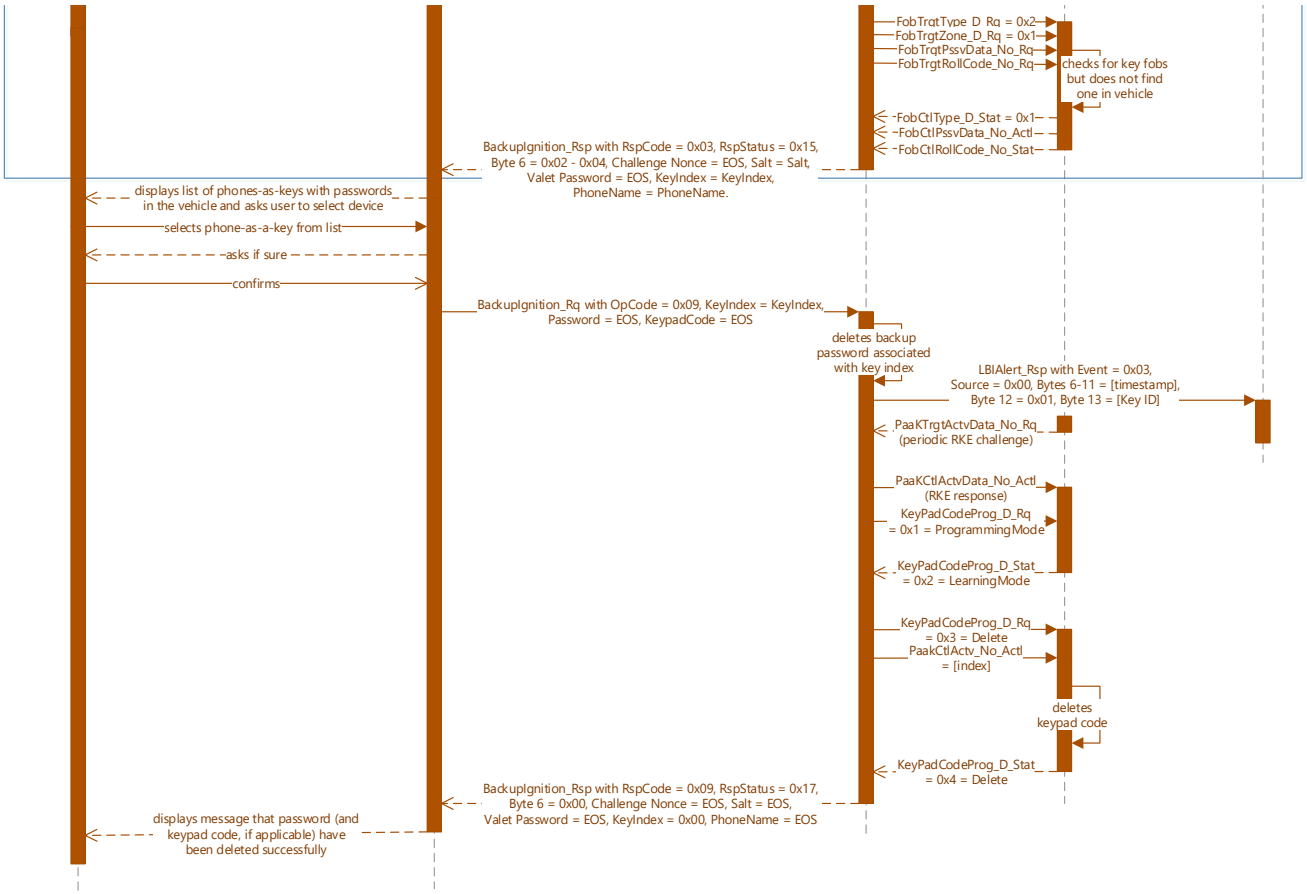
#### *2.1.3.3 Sequence Diagram*

##### 2.1.3.3.1 Deleting backup password for PaaK device

Pre-conditions

1. Vehicle is in RUN and user is inside vehicle.


User
APIM/SYNC
Phone
BLEM
BCM
TCU
selects option to delete backup password for phone-as-a-key
displays requirements to delete backup password
selects to continue
BackupIgnition_Rq with OpCode = 0x03, Byte 5 = 0x00
Phone Not Found
checks for phones-as-keys with passwords but does not find one in vehicle
FobTrqtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrqtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs and finds one in vehicle
FobCtlType_D_Stat = 0x2
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
BackupIgnition_Rsp with RspCode = 0x03, RspStatus = 0x13, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = EOS, PhoneName = EOS
displays message that phone was not found in vehide, asks if they would like to retry or cancel
checks for phones-as-keys with passwords and finds one in vehicle
FobTrqtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrqtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs but does not find one in vehicle
FobCtlType_D_Stat = 0x1
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
BackupIgnition_Rsp with RspCode = 0x03, RspStatus = 0x12, Byte 6 = 0x01, Challenge Nonce = EOS, Salt = Salt, Valet Password = EOS, KeyIndex = KeyIndex, PhoneName = PhoneName
Multiple Phones in Vehicle
checks for phones-as-keys with passwords and finds more than one in vehicle


FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs but does not find one in vehicle
FobCtlType_D_Stat = 0x1
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
BackupIgnition_Rsp with RspCode = 0x03, RspStatus = 0x15, Byte 6 = 0x02 - 0x04, Challenge Nonce = EOS, Salt = Salt, Valet Password = EOS, KeyIndex = KeyIndex, PhoneName = PhoneName.
displays list of phones-as-keys with passwords in the vehicle and asks user to select device
selects phone-as-a-key from list
asks if sure
confirms
BackupIgnition_Rq with OpCode = 0x09, KeyIndex = KeyIndex, Password = EOS, KeypadCode = EOS
deletes backup password associated with key index
LBIAlert_Rsp with Event = 0x03, Source = 0x00, Bytes 6-11 = [timestamp], Byte 12 = 0x01, Byte 13 = [Key ID]
PaaKTrgtActvData_No_Rq (periodic RKE challenge)
PaaKCtlActvData_No_Actl (RKE response)
KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode
KeyPadCodeProg_D_Stat = 0x2 = LearningMode
KeyPadCodeProg_D_Rq = 0x3 = Delete
PaakCtlActv_No_Actl = [index]
deletes keypad code
KeyPadCodeProg_D_Stat = 0x4 = Delete
BackupIgnition_Rsp with RspCode = 0x09, RspStatus = 0x17, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
displays message that password (and keypad code, if applicable) have been deleted successfully

### 2.1.4 Resetting backup password and keypad code for PaaK device

#### *2.1.4.1 Requirements*

**[LBI.R053.02]** When the user selects the option to reset a backup password within SYNC settings, SYNC shall query the BLEM for PaaK devices with passwords in the vehicle and shall request the cryptographic salt (*BackupIgnition_Rq* with *OpCode = 0x03 = Salt and Check for PaaK with Passwords, Byte 5 = 0x00*).

**[LBI.R210.02]** The BLEM shall trigger a BCM Interior Registry search (*FobTrgtType_D_Rq = 0x2 = Registry, FobTrgtZone_D_Rq = 0x1 = Interior, FobTrgtPssvData_No_Rq, FobTrgtRollCode_No_Rq*) whenever it receives *BackupIgnition_Rq* with *OpCode = 0x03 = Salt and Check for PaaK with Passwords*.

***Note: See details of BCM Interior Registry search in Section 3.1.4***

**[LBI.R211.02]** After completing BLEM-requested Interior Registry search (*FobTrgtType_D_Rq = 0x2 = Registry, FobTrgtZone_D_Rq = 0x1 = Interior, FobTrgtPssvData_No_Rq, FobTrgtRollCode_No_Rq*), the BCM shall report to the BLEM whether any key fobs were detected in the vehicle (*FobCtlType_D_Stat, FobCtlPssvData_No_Actl, FobCtlRollCode_No_Stat*).

**[LBI.R054.03]** After receiving Interior Registry search results from the BCM, the BLEM shall report to SYNC what devices (PaaK with passwords or key fob) were found in the vehicle, the names and key indexes of all PaaK devices with passwords found in the vehicle, as well as the cryptographic salt (*BackupIgnition_Rsp* with *RspCode = 0x03 = Salt and Check for PaaK with Passwords Response*).

**[LBI.R055.02]** When SYNC receives *BackupIgnition_Rsp* with *RspCode = 0x03 = Salt and Check for PaaK with Passwords Response*, the SYNC HMI shall display either:

1. Error message that lists phone as missing with options to retry or cancel if:
    - *RspStatus = 0x13 = Fob In Vehicle and No PaaK w/ Password* OR
    - *RspStatus = 0x16 = No PaaK w/o Password and No Fob In Vehicle* AND
    - *Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*
2. List of all detected PaaK devices with instruction for the user to choose the desired device if:
    - *RspStatus = 0x11 = One PaaK w/ Password and Fob In Vehicle* OR
    - *RspStatus = 0x12 = One PaaK w/ Password and No Fob In Vehicle* OR
    - *RspStatus = 0x14 = Two+ PaaK w/ Password and Fob In Vehicle* OR
    - *RspStatus = 0x15 = Two+ PaaK w/ Password and No Fob In Vehicle* AND

- *Byte 6 = 0x01 – 0x04, Challenge Nonce = EOS, Salt = Salt, Valet Password = EOS, KeyIndex = KeyIndex, PhoneName = PhoneName*

**[LBI.R212.02]** When the user chooses their desired device (password) for resetting, the SYNC HMI shall display either:

1. The backup password entry screen (to start Reset Option 1) if:
   - *RspStatus = 0x12 = One PaaK w/ Password and No Fob In Vehicle* OR
   - *RspStatus = 0x15 = Two+ PaaK w/ Password and No Fob In Vehicle*
2. The backup password creation screen (to start Reset Option 2) if:
   - *RspStatus = 0x11 = One PaaK w/ Password and Fob In Vehicle* OR
   - *RspStatus = 0x14 = Two+ PaaK w/ Password and Fob In Vehicle*

*Reset Option 1 starts here.*

**[LBI.R213.02]** When the user enters a password at the backup password entry screen, SYNC shall request a challenge from the BLEM (*BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request, Byte 5 = 0x00*).

**[LBI.R214.01]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request*, it shall issue a challenge to SYNC with cryptographic nonce and salt (*BackupIgnition_Rsp* with *RspCode = 0x01 = Issue Challenge, RspStatus = 0x00 = Reserved, Byte 6 = 0x00, Challenge Nonce = Challenge Nonce, Salt = Salt, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R215.01]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request*, it shall compute, using the cryptographic nonce, another hash of all stored password hashes.

**[LBI.R216.01]** SYNC shall compute a hash of entered password using received salt and then compute a hash of this result using received nonce.

**[LBI.R217.01]** SYNC shall respond to the challenge from the BLEM (*BackupIgnition_Rsp* with *RspCode = 0x01 = Issue Challenge*) with computed password hash (*BackupIgnition_Rq* with *OpCode = 0x0C = Reset Challenge Response, KeyIndex = KeyIndex, Password = Challenge Password, KeypadCode = EOS*).

**[LBI.R218.01]** When the BLEM receives a challenge hash and key index from SYNC (*BackupIgnition_Rq* with *OpCode = 0x0C = Reset Challenge Response, KeyIndex = KeyIndex, Password =*

*Challenge Password, KeypadCode = EOS*), it shall compare the challenge hash with the password hash associated with the received key index.

**[LBI.R219.02]** If the BLEM determines that the received password is invalid i.e. challenge hash does not match password hash associated with received key index, it shall increment invalid password counter and then notify SYNC (*BackupIgnition_Rsp with RspCode = 0x0C = Reset Challenge Response Acknowledge, RspStatus = 0x10 = Invalid Password, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R220.02]** When SYNC receives an invalid password notification (*BackupIgnition_Rsp with RspCode = 0x0C = Reset Challenge Response Acknowledge, RspStatus = 0x10 = Invalid Password*), the SYNC HMI shall notify the user that the entered password is invalid and provide an option to retry.

**[LBI.R221.01]** If the BLEM determines that the received password is valid i.e. challenge hash matches password hash associated with received key index, the SYNC HMI shall display the backup password creation screen.

*Reset Option 2 starts here, and Reset Option 1 continues here.*

**[LBI.R058.01]** The SYNC HMI shall require the user to enter their new backup password twice. If passwords do not match, the SYNC HMI shall notify the user and provide an option to retry.

**[LBI.R059.02]** SYNC shall check entered password against password requirements in real time. The SYNC HMI shall not allow the user to proceed to the next screen until their password meets the minimum requirements.

**[LBI.R060.02]** SYNC shall compute a hash of the entered password and send the result to the BLEM with the key index of the selected device, using either:

1. *OpCode = 0x0D = Reset 1 Password Transmit,* if user is following Reset Option 1
2. *OpCode = 0x0E = Reset 2 Password Transmit,* if user is following Reset Option 2

**[LBI.R222.01]** Upon receiving the password hash and key index, the BLEM shall either:

1. Verify that the PaaK device associated with this key index is still inside the vehicle if *OpCode = 0x0D = Reset 1 Password Transmit*.

2. Verify that the PaaK device associated with this key index AND a key fob are still inside the vehicle if *OpCode = 0x0E = Reset 2 Password Transmit.*

***Note: See details of BCM Interior Registry search in Section 3.1.4***

**[LBI.R223.01]** In Reset Option 1, if the BLEM determines that the PaaK device associated with the received key index is no longer inside the vehicle, then it shall notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x0D = Reset 1 Password Response, RspStatus = 0x07 = PaaK No Longer Detected, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R224.01]** When SYNC receives *BackupIgnition_Rsp* with *RspCode = 0x0D = Reset 1 Password Response, RspStatus = 0x07 = PaaK No Longer Detected*, the SYNC HMI shall display error message that lists phone as missing with options to retry (restart the process) or cancel.

**[LBI.R225.01]** In Reset Option 2, the BLEM shall respond to SYNC with *BackupIgnition_Rsp* with *RspCode = 0x0E = Reset 2 Password Response* and either:

1. *RspStatus = 0x08 = Fob No Longer Detected* if:
    - The BCM does not detect a key fob in the vehicle (*FobCtlType_D_Stat = 0x1 = Invalid*) AND
    - The BLEM detects the PaaK device associated with the received key index.
2. *RspStatus = 0x07 = PaaK No Longer Detected* if:
    - The BCM detects a key fob in the vehicle (*FobCtlType_D_Stat = 0x2 = Valid*) AND
    - The BLEM does not detect the PaaK device associated with the received key index
3. *RspStatus = 0x09 = PaaK and Fob No Longer Detected* if:
    - The BCM does not detect a key fob in the vehicle (*FobCtlType_D_Stat = 0x1 = Invalid*) AND
    - The BLEM does not detect the PaaK device associated with the received key index

**[LBI.R226.01]** In Reset Option 2, when SYNC receives *BackupIgnition_Rsp* with *RspCode = 0x0E = Reset 2 Password Response*, the SYNC HMI shall display either:

1. Error message that lists key fob as missing with options to retry (restart the process) or cancel, if:
    - *RspStatus = 0x08 = Fob No Longer Detected* AND
    - *Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*
2. Error message that lists phone as missing with options to retry (restart the process) or cancel, if:

- *RspStatus = 0x07 = PaaK No Longer Detected* AND
- *Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*

3. Error message that lists key fob and phone as missing with options to retry (restart the process) or cancel, if:
   - *RspStatus = 0x09 = PaaK and Fob No Longer Detected* AND
   - *Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*

**[LBI.R061.02]** The BLEM shall verify that the entered password is not already being used if:

1. In Reset Option 1, the BLEM determines that the PaaK device associated with the received key index is still inside the vehicle.
2. In Reset Option 2, the BLEM determines that the PaaK device associated with the received key index AND a key fob are still inside the vehicle.

**[LBI.R227.01]** If the BLEM determines that the password is already being used, the BLEM shall notify SYNC of this (*BackupIgnition_Rsp* with *RspCode = 0x0D/E = Reset 1/2 Password Response, RspStatus = 0x0A = Password Already Used, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R228.01]** When SYNC receives *BackupIgnition_Rsp* with *RspCode = 0x0D/E = Reset 1/2 Password Response, RspStatus = 0x0A = Password Already Used,* the SYNC HMI shall display a message that password is already being used and instruct user to enter a different password.

**[LBI.R062.03]** If the entered password is not already being used, the BLEM shall then determine whether there is a keypad code associated with the received key index.

If there is an associated keypad code, the BLEM shall initiate keypad code deletion by responding (with *PaaKCtlActvData_No_Actl*) to periodic RKE challenge from the BCM (*PaaKTrgtActvData_No_Rq*) and sending a request to the BCM to enter keypad programming mode (*KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode*).

***Note: See details of BLEM-BCM Keypad programming requirements in Section 3.1.5***

If there is no associated keypad code, the BLEM shall delete the current password hash associated with the received key index.

**[LBI.R063.03]** When the BCM receives *KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode* together with valid RKE response data from the BLEM, it shall enter keypad programming mode and notify the BLEM (*KeyPadCodeProg_D_Stat = 0x2 = LearningMode*) within two seconds and continue to provide notification for up to two seconds.

**[LBI.R064.02]** When the BLEM receives *KeyPadCodeProg_D_Stat = 0x2 = LearningMode*, it shall send to the BCM the received key index (*PaaKCtlActv_No_Actl = [index]*) and a request to delete the personal keypad code (*KeyPadCodeProg_D_Rq = 0x3 = Delete*).

**[LBI.R065.02]** When the BCM receives the request to delete the personal keypad code (*KeyPadCodeProg_D_Rq = 0x3 = Delete*) together with the key index (*PaaKCtlActv_No_Actl = [index]*), it shall delete the personal keypad code that is associated with the received key index.

**[LBI.R230.01]** When the BCM deletes a keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x4 = Delete*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

**[LBI.R066.03]** When the BLEM receives confirmation of keypad code deletion (*KeyPadCodeProg_D_Stat = 0x4 = Delete*), it shall delete the current password hash associated with the received key index.

**[LBI.R337.01]** When the BLEM deletes a backup password hash as part of reset operation, it shall store the new password hash in its HSM and associate it with the received key index.

**[LBI.R338.01]** When the BLEM cannot delete the backup password hash, it shall notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x0D/E = Reset 1/2 Password Response, RspStatus = 0x0C = Password Created Failed, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

*Note: This requirement applies whether there is a keypad code associated with the received key index or not.*

**[LBI.R339.01]** When SYNC receives deletion failure response (*BackupIgnition_Rsp* with *RspCode = 0x0D/E = Reset 1/2 Password Response, RspStatus = 0x0C = Password Created Failed*), the SYNC HMI shall notify user of unsuccessful password creation.

**[LBI.R231.02]** When the BLEM stores a new password hash as part of reset operation, it shall report this to the TCU (*LBIAlert_St* with *Event = 0x09 = Backup Password Reset, Source = 0x00, [timestamp], [Key ID]*) in a BLEM SyncP signed packet (Service Type 0x40/Sub-Service 0x0) as defined in Transfer Protocol BLEM SPSS.

***Note:*** *Reference TP BLEM SPSS and LBI SPSS for BLEM SyncPPacket definition and its payload. Key ID here refers to the Key ID of the PaaK device associated with the backup password that is being reset.*

**[LBI.R232.01]** When the BLEM stores the new password hash, it shall notify SYNC using either:

1. *RspCode = 0x0D = Reset 1 Password Response and RspStatus = 0x0B = Password Created Successfully,* if user is following Reset Option 1
2. *RspCode = 0x0E = Reset 2 Password Response and RspStatus = 0x0B = Password Created Successfully,* if user is following Reset Option 2

**[LBI.R233.01]** When SYNC receives *BackupIgnition_Rsp* with *RspCode = 0x0D/E = Reset 1/2 Password Response, RspStatus = 0x0B = Password Created Successfully*, the SYNC HMI shall notify the user of successful password reset.

**[LBI.R340.01]** When the BCM cannot delete a new keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x5 = ProgrammingFailture*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

**[LBI.R341.01]** When the BLEM receives programming failure response (*KeyPadCodeProg_D_Stat = 0x5 = ProgrammingFailure*), it shall not delete the backup password, but shall notify SYNC of failure (*BackupIgnition_Rsp* with *RspCode = 0x0D/E = Reset 1/2 Password Response, RspStatus = 0x0C = Password Created Failed, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R342.01]** When SYNC receives programming failure response (*BackupIgnition_Rsp* with *RspCode = 0x0D/E = Reset 1/2 Password Response, RspStatus = 0x0C = Password Created Failed*), the SYNC HMI shall notify user of unsuccessful password reset.

**[LBI.R067.02]** After resetting the backup password, the SYNC HMI shall present the user with the option to set up a new personal keypad code.

**[LBI.R068.02]** If the user chooses not to create a personal keypad code, the SYNC HMI shall inform the user that password reset is complete.

**[LBI.R069.02]** If the user chooses to create a personal keypad code, the SYNC HMI shall display a screen for entering new personal keypad code.

*From here, follow keypad code creation process starting* with **LBI.R019** ***and interface details of BLEM-BCM Keypad programming requirements described in Section 3.1.5***

**[LBI.R343.01]** After SYNC notifies user of successful keypad code creation, it shall inform the user that password reset is complete.

#### *2.1.4.2 Use Case*

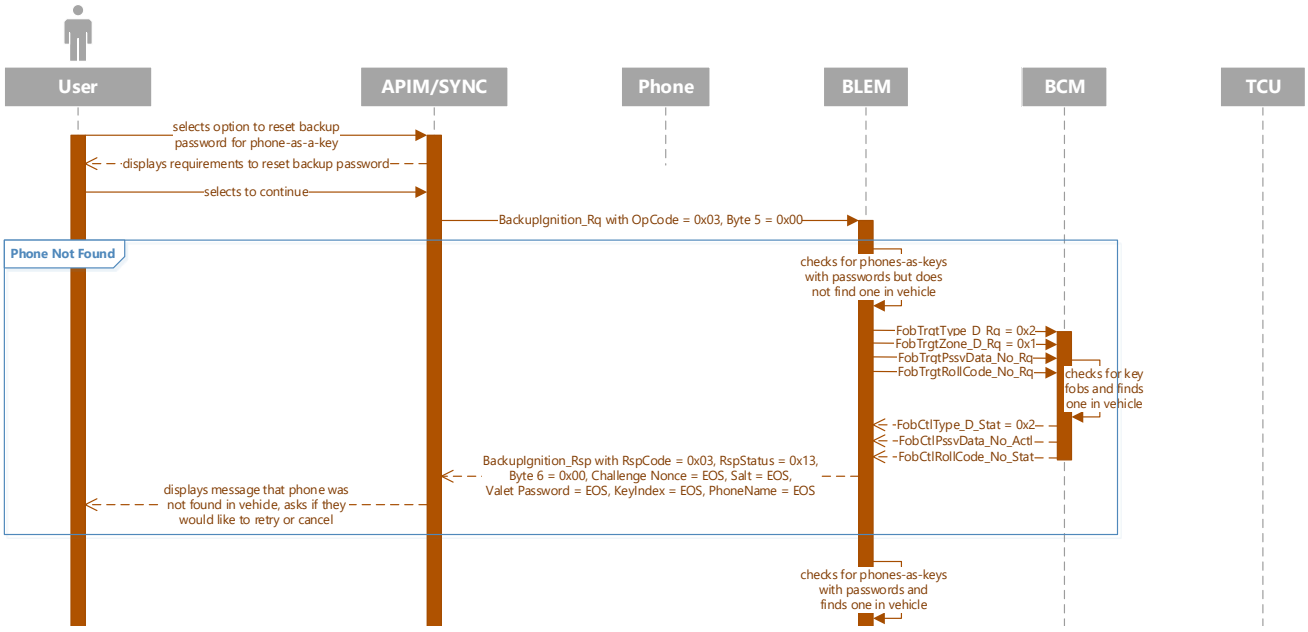
| | |
|---|---|
| **Actors** | User |
| **Pre-conditions** | Vehicle is in RUN.<br>User is inside vehicle.<br>One associated PaaK device with password is inside the vehicle. |
| **Scenario Description** | 1. User selects option to Reset Backup Password for Phone-as-a-Key from PaaK Backup Settings in SYNC.<br>2. SYNC displays message with reset requirements.<br>3. User continues.<br>4. SYNC displays alphanumeric password entry screen and instructs user to enter a backup password.<br>5. User enters password twice according to password requirements.<br>6. User selects Enter.<br>7. SYNC displays message that backup password has been changed successfully and instructs user to commit it to memory. SYNC also asks user if they would like to create a personal keypad code.<br>8. User declines options to create a personal keypad code. |
| **Post-conditions** | Notification that backup password has been deleted is sent to user. |
| **List of Exceptions** | |
| **Interfaces** | APIM<br>BLEM<br>TCU<br>SDN<br>PaaK FI |

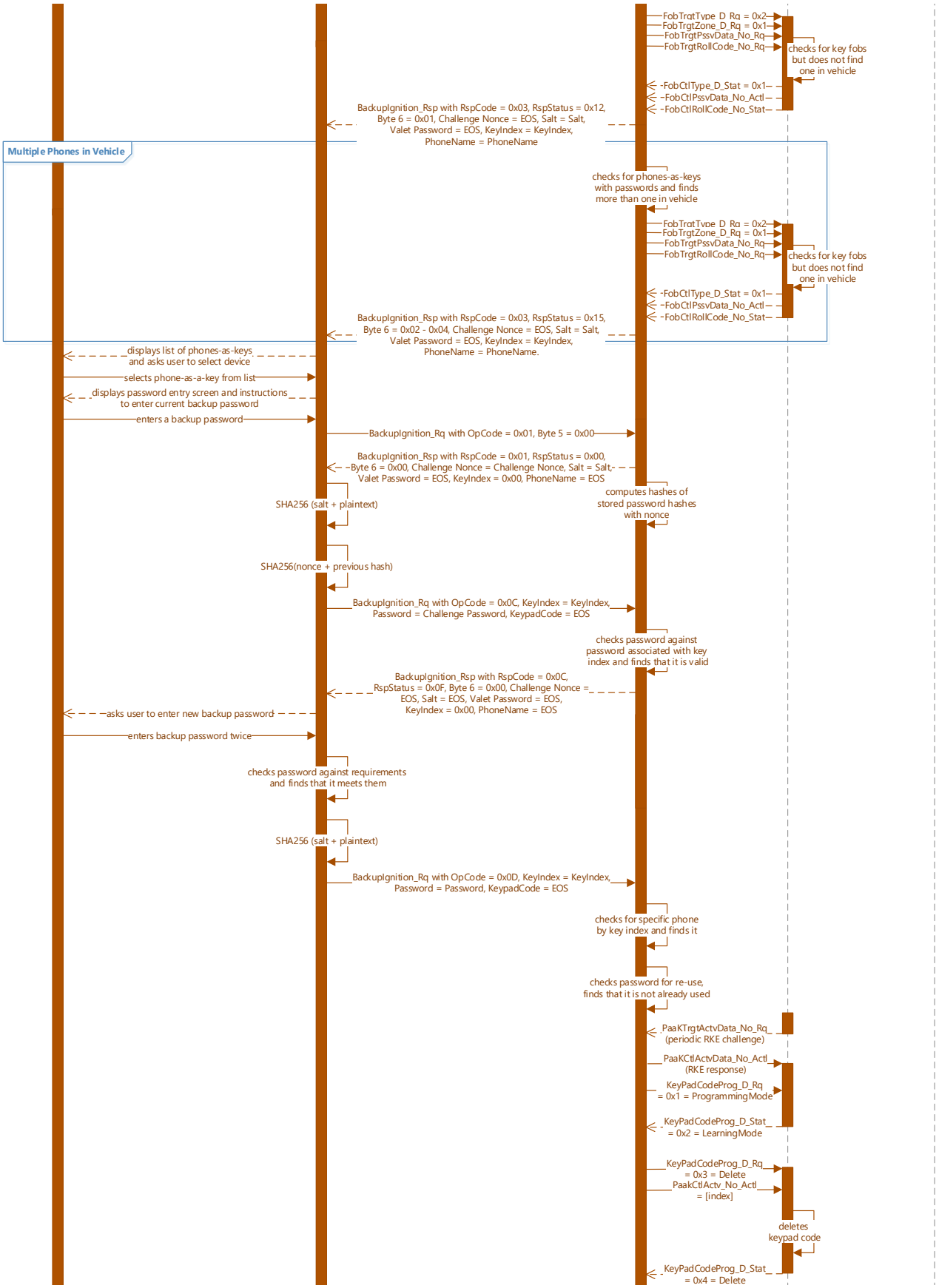
#### *2.1.4.3 Sequence Diagram*

##### 2.1.4.3.1 Resetting backup password for PaaK device – phone and password

Pre-conditions

1. Vehicle is in RUN and user is inside vehicle.

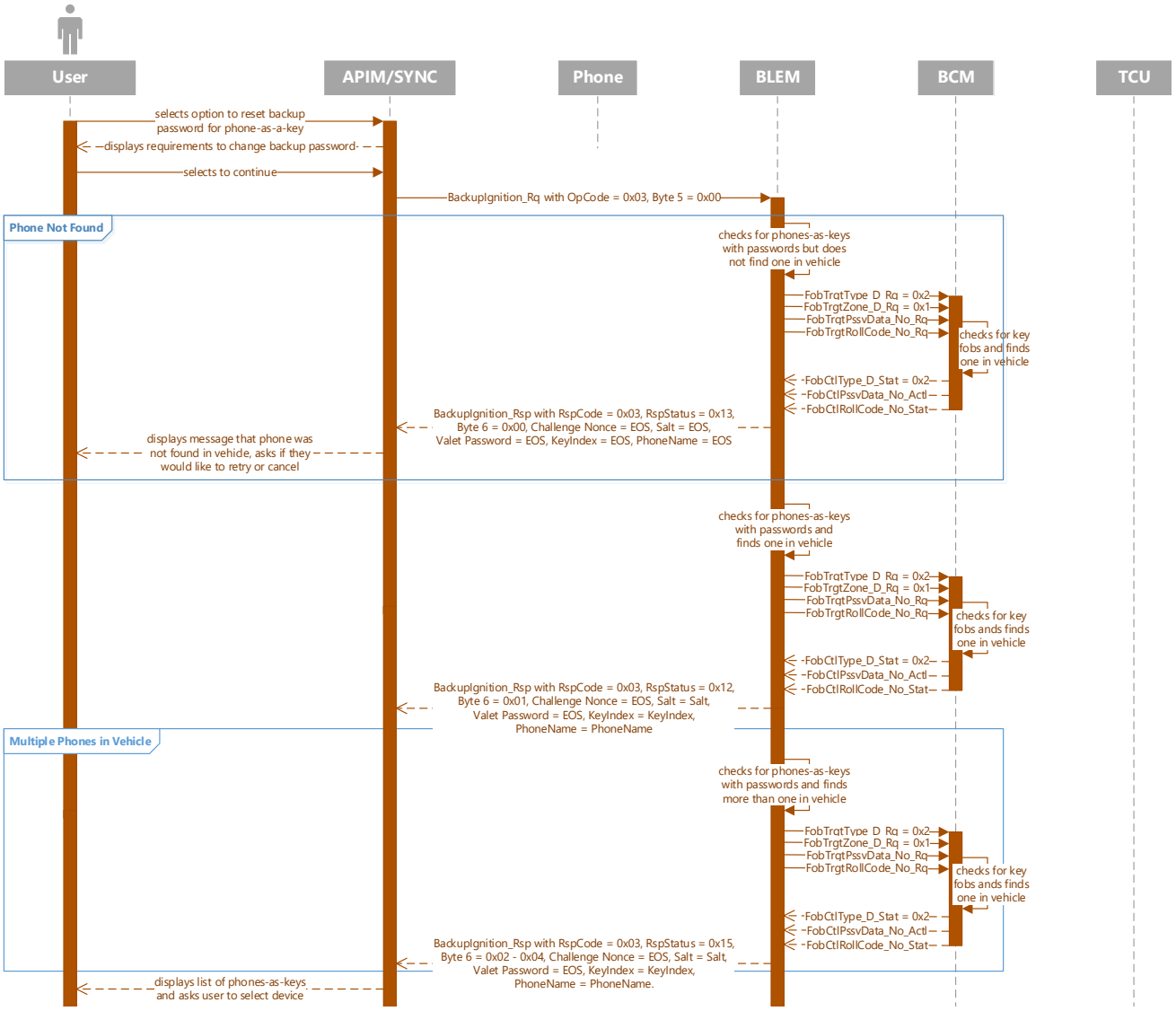


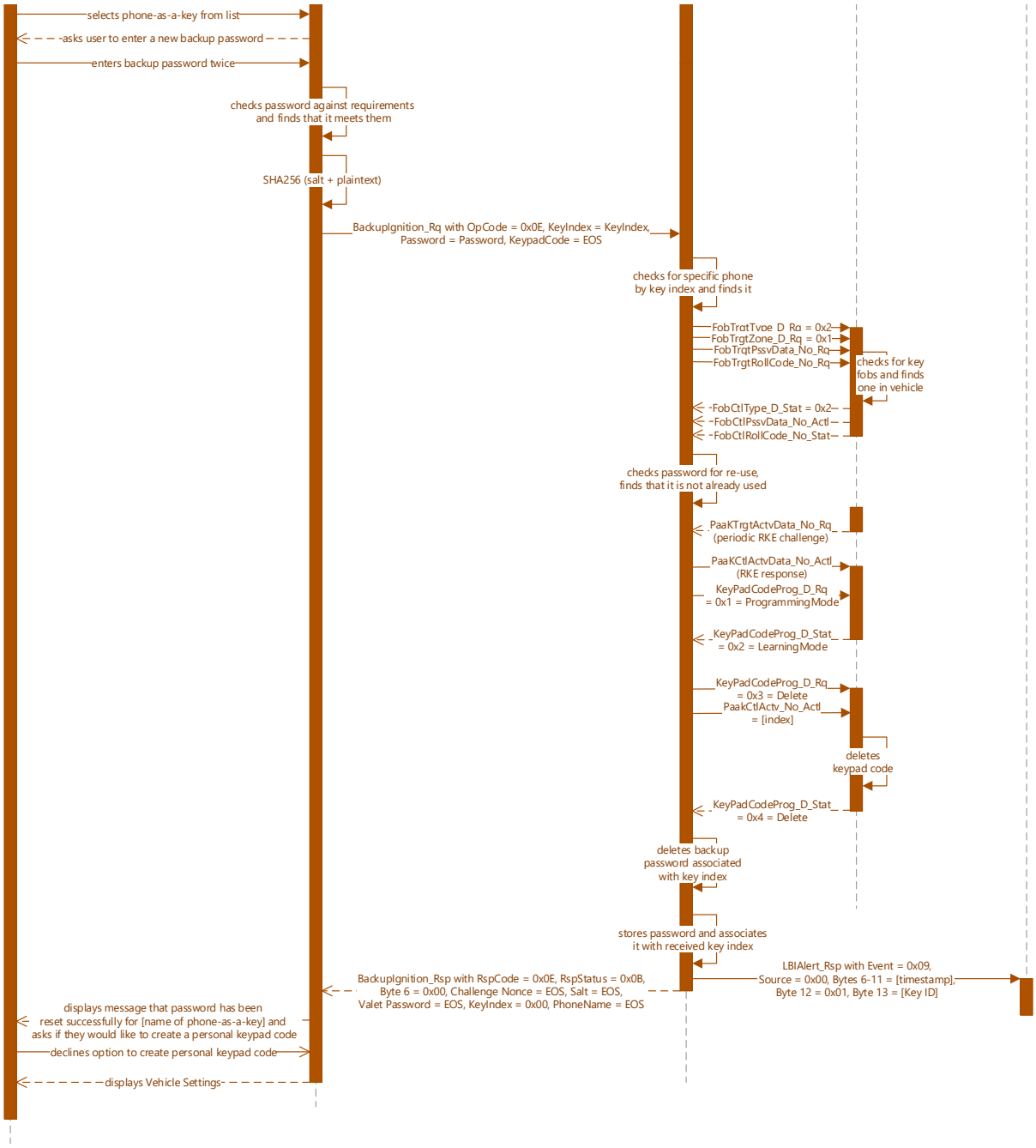

FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs but does not find one in vehicle
FobCtlType_D_Stat = 0x1
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
BackupIgnition_Rsp with RspCode = 0x03, RspStatus = 0x12, Byte 6 = 0x01, Challenge Nonce = EOS, Salt = Salt, Valet Password = EOS, KeyIndex = KeyIndex, PhoneName = PhoneName
Multiple Phones in Vehicle
checks for phones-as-keys with passwords and finds more than one in vehicle
FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs but does not find one in vehicle
FobCtlType_D_Stat = 0x1
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
BackupIgnition_Rsp with RspCode = 0x03, RspStatus = 0x15, Byte 6 = 0x02 - 0x04, Challenge Nonce = EOS, Salt = Salt, Valet Password = EOS, KeyIndex = KeyIndex, PhoneName = PhoneName.
displays list of phones-as-keys and asks user to select device
selects phone-as-a-key from list
displays password entry screen and instructions to enter current backup password
enters a backup password
BackupIgnition_Rq with OpCode = 0x01, Byte 5 = 0x00
BackupIgnition_Rsp with RspCode = 0x01, RspStatus = 0x00, Byte 6 = 0x00, Challenge Nonce = Challenge Nonce, Salt = Salt, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
computes hashes of stored password hashes with nonce
SHA256 (salt + plaintext)
SHA256(nonce + previous hash)
BackupIgnition_Rq with OpCode = 0x0C, KeyIndex = KeyIndex, Password = Challenge Password, KeypadCode = EOS
checks password against password associated with key index and finds that it is valid
BackupIgnition_Rsp with RspCode = 0x0C, RspStatus = 0x0F, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
asks user to enter new backup password
enters backup password twice
checks password against requirements and finds that it meets them
SHA256 (salt + plaintext)
BackupIgnition_Rq with OpCode = 0x0D, KeyIndex = KeyIndex, Password = Password, KeypadCode = EOS
checks for specific phone by key index and finds it
checks password for re-use, finds that it is not already used
PaaKTrgtActvData_No_Rq (periodic RKE challenge)
PaaKCtlActvData_No_Actl (RKE response)
KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode
KeyPadCodeProg_D_Stat = 0x2 = LearningMode
KeyPadCodeProg_D_Rq = 0x3 = Delete
PaakCtlActv_No_Actl = [index]
deletes keypad code
KeyPadCodeProg_D_Stat = 0x4 = Delete


#### 2.1.4.3.2 Resetting backup password for PaaK device – phone and key fob

Pre-conditions

1. Vehicle is in RUN and user is inside vehicle.


User
APIM/SYNC
Phone
BLEM
BCM
TCU
selects option to reset backup password for phone-as-a-key
displays requirements to change backup password
selects to continue
BackupIgnition_Rq with OpCode = 0x03, Byte 5 = 0x00
Phone Not Found
checks for phones-as-keys with passwords but does not find one in vehicle
FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs and finds one in vehicle
FobCtlType_D_Stat = 0x2
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
BackupIgnition_Rsp with RspCode = 0x03, RspStatus = 0x13, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = EOS, PhoneName = EOS
displays message that phone was not found in vehide, asks if they would like to retry or cancel
checks for phones-as-keys with passwords and finds one in vehicle
FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs ands finds one in vehicle
FobCtlType_D_Stat = 0x2
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
BackupIgnition_Rsp with RspCode = 0x03, RspStatus = 0x12, Byte 6 = 0x01, Challenge Nonce = EOS, Salt = Salt, Valet Password = EOS, KeyIndex = KeyIndex, PhoneName = PhoneName
Multiple Phones in Vehicle
checks for phones-as-keys with passwords and finds more than one in vehicle
FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs ands finds one in vehicle
FobCtlType_D_Stat = 0x2
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
BackupIgnition_Rsp with RspCode = 0x03, RspStatus = 0x15, Byte 6 = 0x02 - 0x04, Challenge Nonce = EOS, Salt = Salt, Valet Password = EOS, KeyIndex = KeyIndex, PhoneName = PhoneName.
displays list of phones-as-keys and asks user to select device


selects phone-as-a-key from list
asks user to enter a new backup password
enters backup password twice
checks password against requirements and finds that it meets them
SHA256 (salt + plaintext)
BackupIgnition_Rq with OpCode = 0x0E, KeyIndex = KeyIndex, Password = Password, KeypadCode = EOS
checks for specific phone by key index and finds it
FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs and finds one in vehicle
FobCtlType_D_Stat = 0x2
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
checks password for re-use, finds that it is not already used
PaaKTrgtActvData_No_Rq (periodic RKE challenge)
PaaKCtlActvData_No_Actl (RKE response)
KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode
KeyPadCodeProg_D_Stat = 0x2 = LearningMode
KeyPadCodeProg_D_Rq = 0x3 = Delete
PaakCtlActv_No_Actl = [index]
deletes keypad code
KeyPadCodeProg_D_Stat = 0x4 = Delete
deletes backup password associated with key index
stores password and associates it with received key index
LBIAlert_Rsp with Event = 0x09, Source = 0x00, Bytes 6-11 = [timestamp], Byte 12 = 0x01, Byte 13 = [Key ID]
BackupIgnition_Rsp with RspCode = 0x0E, RspStatus = 0x0B, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
displays message that password has been reset successfully for [name of phone-as-a-key] and asks if they would like to create a personal keypad code
declines option to create personal keypad code
displays Vehicle Settings

### 2.1.5 Generating valet password and keypad code

#### *2.1.5.1 Requirements*

**[LBI.R077.03]** SYNC shall allow access to Enhanced Valet Mode whenever a backup password is stored in the BLEM (*IgnPsswrdActv_B_Stat = 0x1 = Active*). When *IgnPsswrdActv_B_Stat = 0x1 = Active* and the user selects the Valet Mode button in the SYNC HMI, the vehicle shall initiate either the existing Valet Mode or Enhanced Valet Mode experience, depending on which devices are present in the vehicle.

**[LBI.R078.03]** When *IgnPsswrdActv_B_Stat = 0x1 = Active* and the user selects the Valet Mode button in the SYNC HMI, SYNC shall query the BLEM for PaaK devices in the vehicle (*BackupIgnition_Rq* with *OpCode = 0x05 = Check for Keys to Enter Valet Mode, Byte 5 = 0x00*).

**[LBI.R235.02]** The BLEM shall trigger a BCM Interior Registry search (*FobTrgtType_D_Rq = 0x2 = Registry, FobTrgtZone_D_Rq = 0x1 = Interior, FobTrgtPssvData_No_Rq, FobTrgtRollCode_No_Rq*) whenever it receives *BackupIgnition_Rq* with *OpCode = 0x05 = Check for Keys to Enter Valet Mode*.

***Note: See details of BCM Interior Registry search in Section 3.1.4***

**[LBI.R236.02]** After completing BLEM-requested Interior Registry search (*FobTrgtType_D_Rq = 0x2 = Registry, FobTrgtZone_D_Rq = 0x1 = Interior, FobTrgtPssvData_No_Rq, FobTrgtRollCode_No_Rq*), the BCM shall report to the BLEM whether any key fobs were detected in the vehicle (*FobCtlType_D_Stat, FobCtlPssvData_No_Actl, FobCtlRollCode_No_Stat*).

**[LBI.R237.03]** After receiving Interior Registry search results from the BCM, the BLEM shall verify vehicle configuration and either:

1. Notify SYNC, if a key fob is detected regardless of whether a PaaK device is detected inside the vehicle (to start Existing Valet Mode).
    - *BackupIgnition_Rsp* with *RspCode = 0x05 = Check for Keys to Enter Valet Mode, RspStatus = 0x1B = Fob In Vehicle, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*
2. Generate a random 8-digit number (when vehicle configuration requires the use of 5-digit keypad codes) or generate a random 10-digit number (when vehicle configuration requires the use of 7-digit keypad codes), if a key fob is not detected, but at least one PaaK device is detected inside the vehicle (to start Activating Enhanced Valet Mode Option 1).
3. Notify SYNC, if neither a PaaK device nor a key fob are detected inside the vehicle (to start Activating Enhanced Valet Mode Option 2)

- *BackupIgnition_Rsp* with *RspCode = 0x05 = Check for Keys to Enter Valet Mode, RspStatus = 0x1C = No PaaK and No Fob In Vehicle, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*

*Existing Valet Mode starts here.*

**[LBI.R079.03]** When SYNC receives *BackupIgnition_Rsp* with *RspCode = 0x05 = Check for Keys to Enter Valet Mode* and *RspStatus = 0x1B = Fob In Vehicle*, the SYNC HMI shall display existing Valet Mode (PIN entry screen).

*Activating Enhanced Valet Mode Option 1 starts here.*

**[LBI.R238.03]** When the BLEM generates a random 8-digit or 10-digit number, it shall store in NVM the key indexes of all PaaK devices that were detected inside the vehicle during SYNC-requested key search (*BackupIgnition_Rq* with *OpCode = 0x05 = Check for Keys to Enter Valet Mode, Byte 5 = 0x00*).

**[LBI.R081.04]** After the BLEM generates the random 8-digit or 10-digit number, it shall initiate valet keypad code storage by responding to periodic RKE challenge from the BCM (*PaaKTrgtActvData_No_Rq*) and send a request to the BCM to enter keypad programming mode (*KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode*).

***Note: See details of BLEM-BCM Keypad programming requirements in Section 3.1.5***

**[LBI.R082.03]** When the BCM receives *KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode* together with valid RKE response data from the BLEM, it shall enter keypad programming mode and notify the BLEM (*KeyPadCodeProg_D_Stat = 0x2 = LearningMode*) within two seconds and continue to provide notification for up to two seconds.

**[LBI.R083.03]** When the BLEM receives *KeyPadCodeProg_D_Stat = 0x2 = LearningMode*, it shall send to the BCM the first five digits of the random 8-digit number (valet keypad code) when vehicle configuration requires the use of the 5-digit keypad codes or the first seven digits of the random 10-digit number (valet keypad code) when vehicle configuration requires the use of the 7-digit keypad codes (in *PaaKCtlActvData_No_Actl*), the valet key index (*PaaKCtlActv_No_Actl = 63*) and a request to store the valet keypad code (*KeyPadCodeProg_D_Rq = 0x2 = Add*).

**[LBI.R084.02]** When the BCM receives the request to store the valet keypad code (*KeyPadCodeProg_D_Rq = 0x2 = Add*) together with the valet key index (*PaaKCtlActv_No_Actl = 63*) and the keypad code (in *PaaKCtlActvData_No_Actl*), it shall store the received keypad code and associate it with key index 63.

**[LBI.R239.01]** When the BCM stores a new keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x3 = Add*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

**[LBI.R312.01]** When the BCM cannot store a new keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x5 = ProgrammingFailture*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

**[LBI.R313.01]** When the BLEM receives programming failure response (*KeyPadCodeProg_D_Stat = 0x5 = ProgrammingFailure*), it shall notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x05 = Check for Keys to Enter Valet Mode, RspStatus = 0x0C = Password Created Failed, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R314.01]** When SYNC receives programming failure response (*BackupIgnition_Rsp* with *RspCode = 0x05 = Check for Keys to Enter Valet Mode, RspStatus = 0x0C = Password Created Failed*), the SYNC HMI shall notify user of programming failure.

**[LBI.R085.02]** When the BLEM receives confirmation of valet keypad code storage (*KeyPadCodeProg_D_Stat = 0x3 = Add*), it shall store the valet password hash and then associate this password entry with key index 63 in its HSM.

**[LBI.R240.02]** Whenever an active password entry is associated with key index 63, the BLEM shall recognize this as being in Enhanced Valet Mode.

**[LBI.R241.02]** When the BLEM stores a new valet password hash, it shall report this to the TCU (*LBIAlert_St* with *Event = 0x04 = Valet Password Created, Source = 0x02 = PaaK, [timestamp], [Key ID]*) in a BLEM SyncP signed packet (Service Type 0x40/Sub-Service 0x0) as defined in Transfer Protocol BLEM SPSS.

***Note:*** *Reference TP BLEM SPSS and LBI SPSS for BLEM SyncPPacket definition and its payload. Key ID here refers to the Key IDs of the PaaK devices that were in the vehicle at the time of valet password generation.*

**[LBI.R242.02]** When the BLEM stores a new valet password hash, it shall send the valet password (plain text) via encrypted PaaK BLE interface to all PaaK devices that were detected in

the vehicle during SYNC-requested key search. The valet password shall be contained within the payload (defined below) of a GATT write command (reference Phone as a Key BLE Communication Protocol).

| Message Parameters | Value | Size | Description |
|---|---|---|---|
| Command | 0x15 | 1 Byte | Write valet password command |
| Payload | 0x00000000 – 0x05F5E0FF | 4 Bytes | 8-digit numeric password represented as an unsigned integer |
| Key Index | 0x00 – 0x3F | 1 Byte | Key index of device to which password will be sent. Used as identification for return status. |
| Padding | 0x0202 | 2 Bytes | Padding following requirements in Phone as a Key BLE Communication Protocol, Section 9 |

**[LBI.R086.03]** The BLEM shall wait for up to one second for at least one PaaK device to confirm password delivery. If one PaaK device confirms password delivery (i.e. Success, according to table below), BLEM shall immediately send confirmation of this fact with the valet password (plain text) to SYNC (*BackupIgnition_Rsp* with *RspCode = 0x05 = Check for Keys to Enter Valet Mode, RspStatus = 0x1D = Password Created Successfully and Delivered to PaaK, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = Valet Password, KeyIndex = 0x00, PhoneName = EOS*).

| Message Parameters | Value | Size | Description |
|---|---|---|---|
| Command | 0x16 | 1 Byte | Write valet password acknowledge |
| Status | 0x32 – 0x33 | 1 Byte | Success (0x32) means PaaK device received password. Failure (0x33) means PaaK device received password, but could not process it. |
| Key Index | 0x00 – 0x3F | 1 Byte | Key index of device providing status. |
| Padding | 0x0505050505 | 5 Bytes | Padding following requirements in Phone as a Key BLE Communication Protocol, Section 9 |

**[LBI.R344.01]** If no PaaK devices have confirmed password delivery at the end of one second, the BLEM shall send the valet password (plain text) to SYNC (*BackupIgnition_Rsp* with *RspCode = 0x05 = Check for Keys to Enter Valet Mode, RspStatus = 0x0B = Password Created Successfully, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = Valet Password, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R243.01]** Once the BLEM sends the plain text valet password to SYNC, it shall delete it from memory.

**[LBI.R345.01]** When SYNC receives from the BLEM the valet password and confirmation of delivery to PaaK device (*BackupIgnition_Rsp* with *RspCode = 0x05 = Check for Keys to Enter Valet Mode, RspStatus = 0x1D, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = Valet Password, KeyIndex = 0x00, PhoneName = EOS*), the SYNC HMI shall display a message stating that valet password has been delivered to PaaK device and then enter Enhanced Valet Mode.

**[LBI.R087.03]** When SYNC receives from the BLEM the valet password but no confirmation of delivery to PaaK device (*BackupIgnition_Rsp* with *RspCode = 0x05 = Check for Keys to Enter Valet Mode, RspStatus = 0x0B, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = Valet Password, KeyIndex = 0x00, PhoneName = EOS*), the SYNC HMI shall enter Enhanced Valet Mode.

*Activating Enhanced Valet Mode Option 2 starts here.*

**[LBI.R088.03]** When SYNC receives *BackupIgnition_Rsp* with *RspCode = 0x05 = Check for Keys to Enter Valet Mode, RspStatus = 0x1C = No PaaK and No Fob In Vehicle*, the SYNC HMI shall display the backup password entry screen and ask the user to enter a backup password.

**[LBI.R244.02]** When the user enters a password at the backup password entry screen, SYNC shall request a challenge from the BLEM (*BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request, Byte 5 = 0x00*).

**[LBI.R245.01]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request*, it shall issue a challenge to SYNC with cryptographic nonce and salt (*BackupIgnition_Rsp* with *RspCode = 0x01 = Issue Challenge, RspStatus = 0x00 = Reserved, Byte 6 = 0x00, Challenge Nonce = Challenge Nonce, Salt = Salt, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R246.01]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request*, it shall compute, using the cryptographic nonce, another hash of all stored password hashes.

**[LBI.R091.02]** SYNC shall compute a hash of entered password using received salt and then compute a hash of this result using received nonce.

**[LBI.R092.02]** SYNC shall respond to the challenge from the BLEM (*BackupIgnition_Rsp* with *RspCode = 0x01 = Issue Challenge*) with computed password hash (*BackupIgnition_Rq* with *OpCode = 0x0A = Valet Create Challenge Response, KeyIndex = 0x00, Password = Challenge Password, KeypadCode = EOS*).

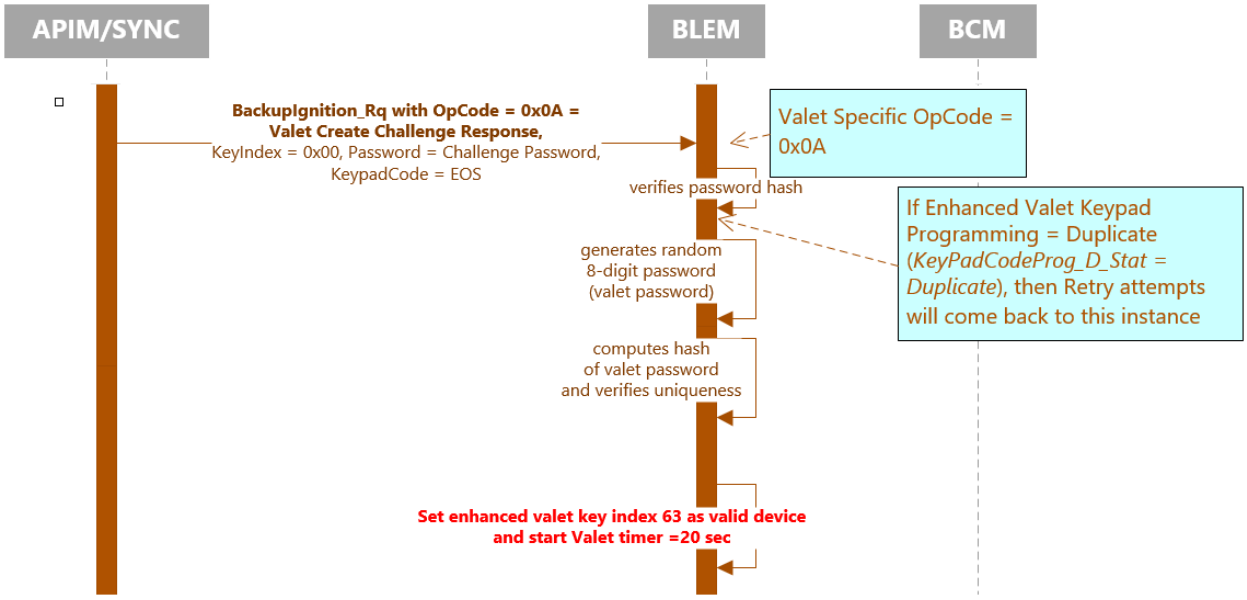
**[LBI.R093.03]** When the BLEM receives a challenge hash from SYNC (*BackupIgnition_Rq* with *OpCode = 0x0A = Valet Create Challenge Response, KeyIndex = 0x00, Password = Challenge Password, KeypadCode = EOS*), it shall compare it with the hashes that it computed for the stored passwords.

If the BLEM determines that the challenge hash matches a calculated password hash it shall initiate a valet timer, $Valet_{timer}$ = 20 sec.

When the BLEM is in a 20-second timer period while *Ignition_Status = Run*, the BLEM shall respond positively (*PaakCtlType_D_Stat = 0x2 = Valid, PaakCtlIndx1_No_Actl = [63]*) to BCM Crypto start searches (*PaakTrgtType_D_Rq = 0x1 = Crypto, PaakTrgtZone_D_Rq = 0x1 = Interior)*.

After this timer expires, the BLEM shall respond negatively (*PaakCtlType_D_Stat = 0x1 = Invalid*) to BCM Crypto Start searches when PaaK device has not been detected in vehicle interior zone.

See below diagram:




The BLEM shall de-activate the timer once the BCM clears search criteria (Type/Zone = Null/Null) or if the BLEM detects the Ignition status changed from Run to Off.

***Note: "Clears search criteria" means that search request has been received, the BLEM has responded, and the BCM stops the active search.***

If Enhanced valet keypad programming status reported by BCM to be duplicate, *KeyPadCodeProg_D_Stat = Duplicate,* then the BLEM shall retry to re-generate Enhanced Valet password at the instance shown in the above diagram.

**[LBI.R094.03]** If the BLEM determines that the received password is invalid i.e. challenge hash does not match a calculated password hash, it shall increment invalid password counter and then notify SYNC (*BackupIgnition_Rsp with RspCode = 0x0A = Valet Create Challenge Response Acknowledge, RspStatus = 0x10 = Invalid Password, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R095.03]** When SYNC receives an invalid password notification (*BackupIgnition_Rsp with RspCode = 0x0A = Valet Create Challenge Response Acknowledge, RspStatus = 0x10 = Invalid Password*), the SYNC HMI shall notify the user that the entered password is invalid and provide an option to retry.

**[LBI.R096.02]** If the BLEM determines that the received password is valid i.e. challenge hash matches a calculated password hash, it shall generate a random 8-digit number.

**[LBI.R097.04]** After the BLEM generates the random 8-digit or 10-digit number (based on vehicle configuration), it shall initiate valet keypad code storage by responding to periodic RKE challenge from the BCM (*PaaKTrgtActvData_No_Rq*) and sending a request to the BCM to enter keypad programming mode (*KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode*).

***Note: See details of BLEM-BCM Keypad programming requirements in Section 3.1.5***

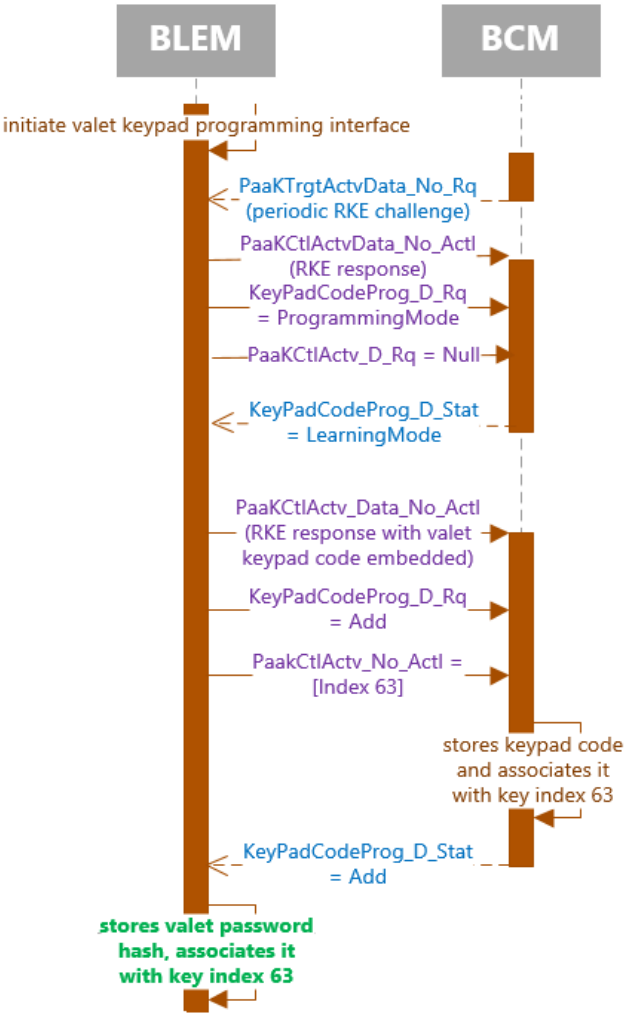
**[LBI.R098.02]** When the BCM receives *KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode* together with valid RKE response data from the BLEM, it shall enter keypad programming mode and notify the BLEM (*KeyPadCodeProg_D_Stat = 0x2 = LearningMode*) within two seconds and continue to provide notification for two seconds.

**[LBI.R099.02]** When the BLEM receives *KeyPadCodeProg_D_Stat = 0x2 = LearningMode*, it shall send to the BCM the first five digits of the random 8-digit number (valet keypad code) (in *PaaKCtlActvData_No_Actl*), the valet key index (*PaaKCtlActv_No_Actl = 63*) and a request to store the valet keypad code (*KeyPadCodeProg_D_Rq = 0x2 = Add*).

**[LBI.R100.02]** When the BCM receives the request to store the valet keypad code (*KeyPadCodeProg_D_Rq = 0x2 = Add*) together with the valet key index (*PaaKCtlActv_No_Actl = 63*) and the keypad code (in *PaaKCtlActvData_No_Actl*), it shall store the received keypad code and associate it with key index 63.

**[LBI.R247.01]** When the BCM stores a new keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x3 = Add*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

**[LBI.R101.01]** When the BLEM receives confirmation of valet keypad code storage (*KeyPadCodeProg_D_Stat = 0x3 = Add*), it shall compute a hash of the valet password and then associate this password entry with key index 63 in its HSM.
See below diagram:



**[LBI.R248.03]** When the BLEM stores a new valet password hash, it shall report this to the TCU (*LBIAlert_St* with *Event = 0x04 = Valet Password Created, Source = 0x01 = Password, [timestamp], [Key ID]*) in a BLEM SyncP signed packet (Service Type 0x40/Sub-Service 0x0) as defined in Transfer Protocol BLEM SPSS.

***Note:*** *Reference TP BLEM SPSS and LBI SPSS for BLEM SyncPPacket definition and its payload. Key ID here refers to the Key ID of the PaaK device associated with the entered backup password.*

**[LBI.R249.02]** After the BLEM stores a new valet password hash, it shall send the valet password (plain text) to SYNC (*BackupIgnition_Rsp* with *RspCode = 0x0A = Valet Create Challenge Response, RspStatus = 0x0B = Password Created Successfully, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = Valet Password, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R102.03]** After SYNC receives the valet password from the BLEM (*BackupIgnition_Rsp* with *RspCode = 0x0A = Valet Create Challenge Response, RspStatus = 0x0B, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = Valet Password, KeyIndex = 0x00, PhoneName = EOS*), the SYNC HMI shall enter Enhanced Valet Mode.

**[LBI.R346.01]** When SYNC is in Enhanced Valet Mode, it shall apply the same restrictions as existing Valet Mode, meaning the user interface and the following functions are restricted:

- Navigation
- USB
- CD State
- Emergency assistance
- Audio
- Bluetooth
- Sirius
- Multimedia

**[LBI.R250.02]** SYNC shall hold the status (i.e. active, inactive) of Enhanced Valet Mode in NVM.

#### *2.1.5.2 Use Case*

| | |
|---|---|
| **Actors** | User |
| **Pre-conditions** | User has previously created a backup password.<br>Vehicle is in RUN. |

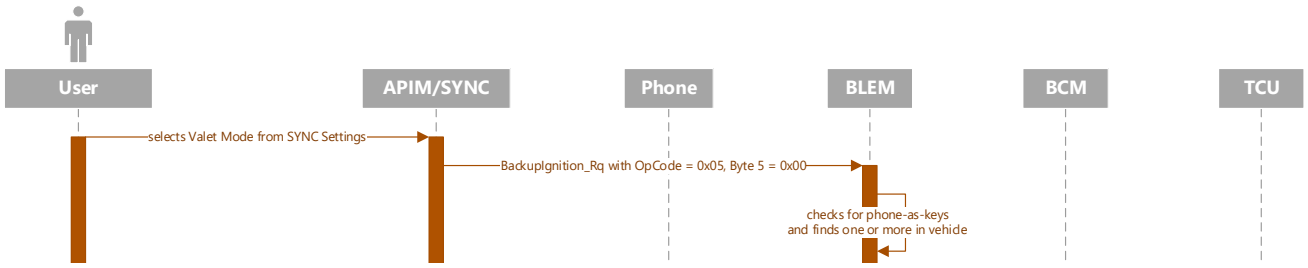
| | |
|---|---|
| | User is inside vehicle.<br>One associated PaaK device is inside the vehicle. |
| **Scenario Description** | 1. User selects Valet Mode from SYNC Settings.<br>2. PaaK device receives notification with valet password.<br>3. SYNC displays message that valet password has been sent to connected device.<br>4. SYNC displays screen that says Enhanced Valet Mode is active, shows valet password, and explains how to use password. |
| **Post-conditions** | Valet password will be displayed in SYNC until vehicle is shutdown.<br>Notification that valet password has been created is sent to user. |
| **List of Exception Use Cases** | |
| **Interfaces** | APIM<br>BCM<br>BLEM<br>PaaK device<br>TCU<br>SDN<br>PaaK FI |

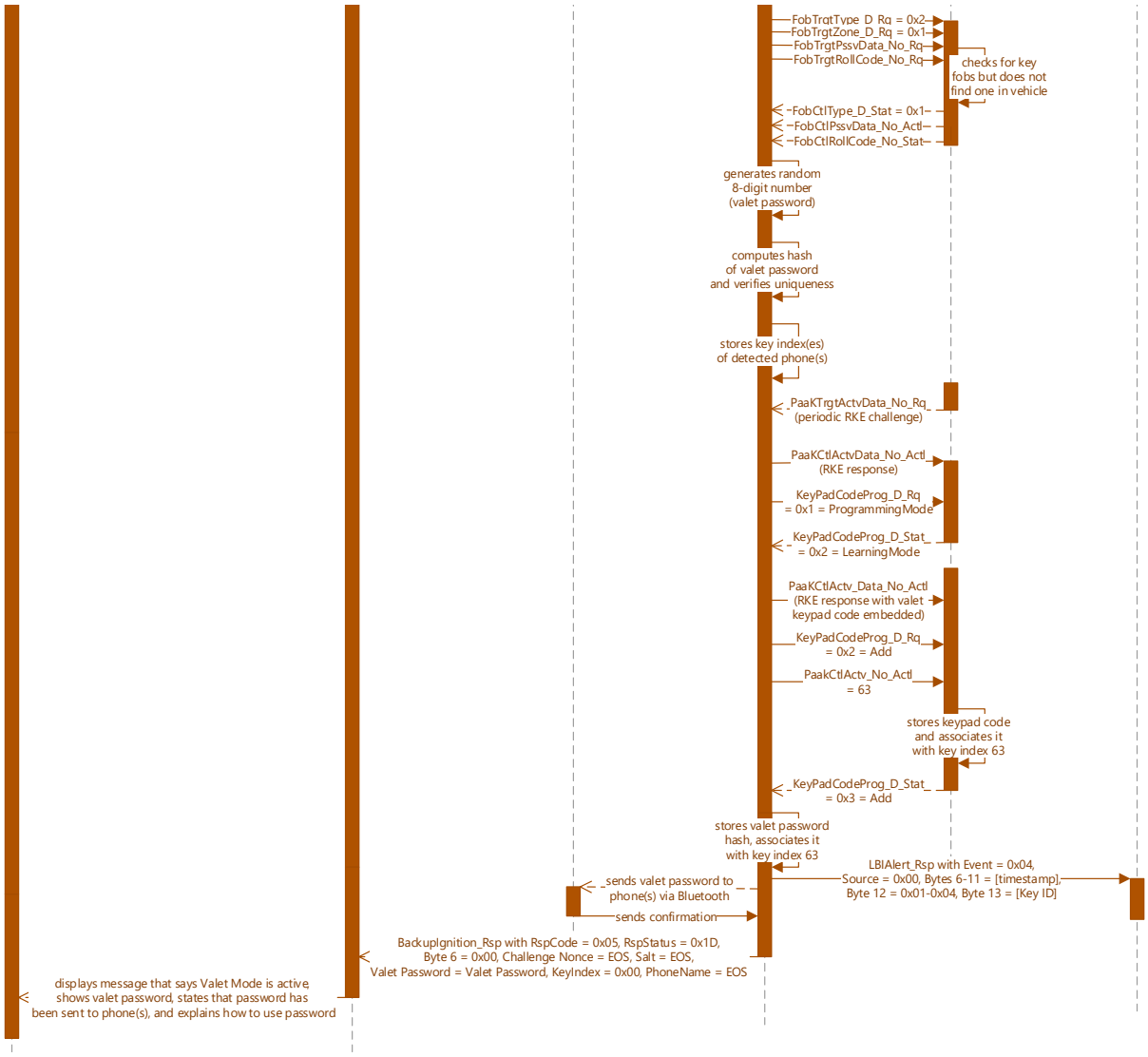
#### *2.1.5.3 Sequence Diagrams*

##### 2.1.5.3.1 Generating valet password and keypad code – phone(s) present without key fob

Pre-conditions

1. User has previously created backup password.
2. Vehicle is in RUN and user is inside vehicle.
3. BLEM has vehicle-unique salt.

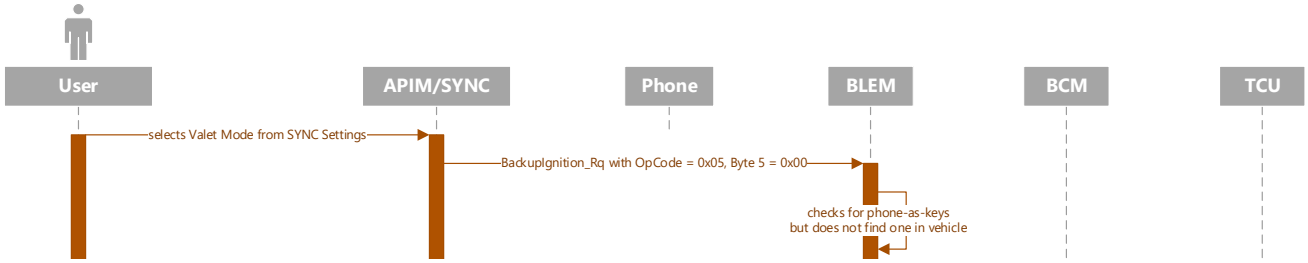


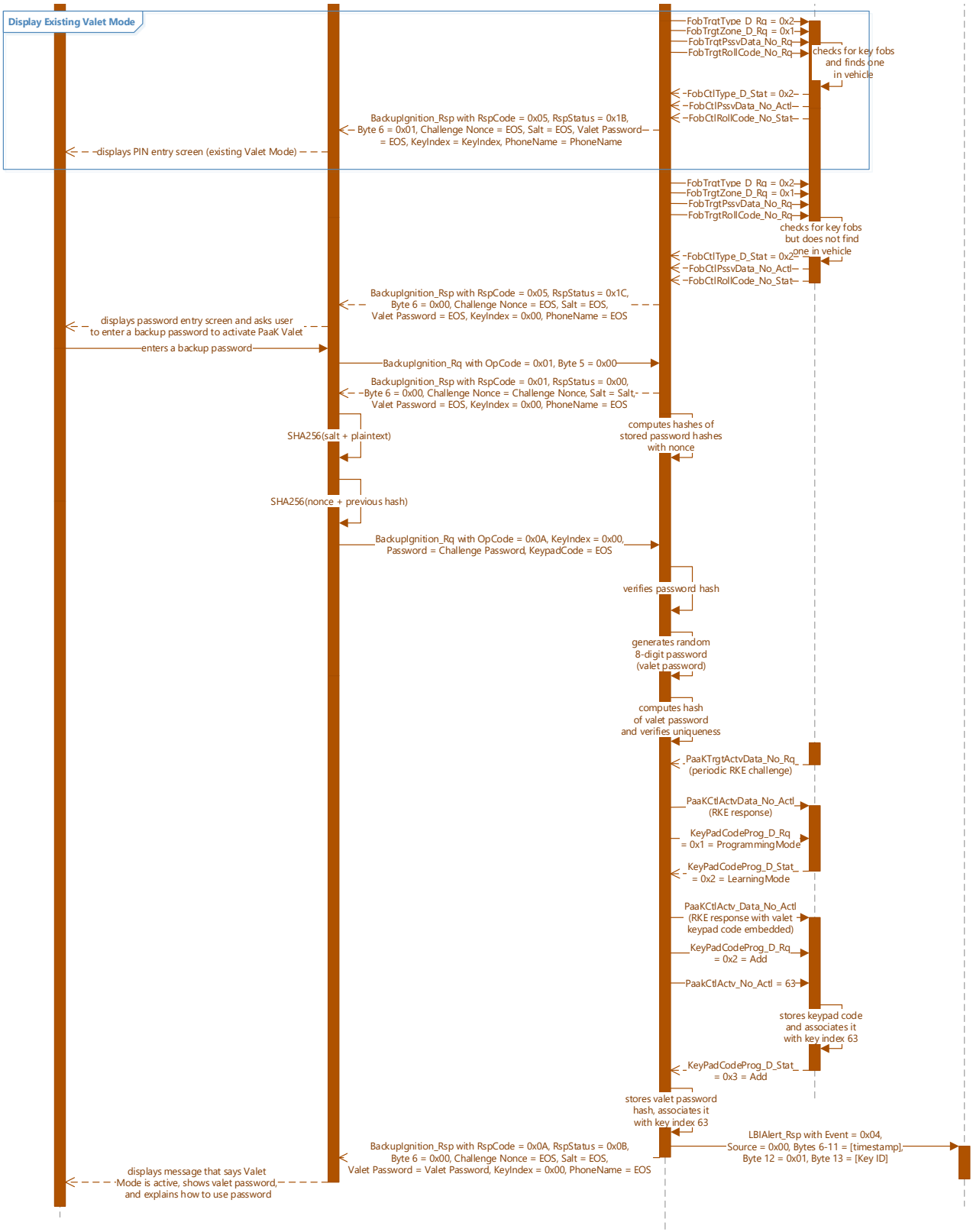



#### 2.1.5.3.2 Generating valet password and keypad code – no devices present

Pre-conditions

1. User has previously created backup password.
2. Vehicle is in RUN and user is inside vehicle.
3. BLEM has vehicle-unique salt.


User
APIM/SYNC
Phone
BLEM
BCM
TCU
selects Valet Mode from SYNC Settings
BackupIgnition_Rq with OpCode = 0x05, Byte 5 = 0x00
checks for phone-as-keys
but does not find one in vehicle


Display Existing Valet Mode
FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs and finds one in vehicle
FobCtlType_D_Stat = 0x2
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
BackupIgnition_Rsp with RspCode = 0x05, RspStatus = 0x1B, Byte 6 = 0x01, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = KeyIndex, PhoneName = PhoneName
displays PIN entry screen (existing Valet Mode)
FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs but does not find one in vehicle
FobCtlType_D_Stat = 0x2
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
BackupIgnition_Rsp with RspCode = 0x05, RspStatus = 0x1C, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
displays password entry screen and asks user to enter a backup password to activate PaaK Valet
enters a backup password
BackupIgnition_Rq with OpCode = 0x01, Byte 5 = 0x00
BackupIgnition_Rsp with RspCode = 0x01, RspStatus = 0x00, Byte 6 = 0x00, Challenge Nonce = Challenge Nonce, Salt = Salt, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
computes hashes of stored password hashes with nonce
SHA256(salt + plaintext)
SHA256(nonce + previous hash)
BackupIgnition_Rq with OpCode = 0x0A, KeyIndex = 0x00, Password = Challenge Password, KeypadCode = EOS
verifies password hash
generates random 8-digit password (valet password)
computes hash of valet password and verifies uniqueness
PaaKTrgtActvData_No_Rq (periodic RKE challenge)
PaaKCtlActvData_No_Actl (RKE response)
KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode
KeyPadCodeProg_D_Stat = 0x2 = LearningMode
PaaKCtlActv_Data_No_Actl (RKE response with valet keypad code embedded)
KeyPadCodeProg_D_Rq = 0x2 = Add
PaakCtlActv_No_Actl = 63
stores keypad code and associates it with key index 63
KeyPadCodeProg_D_Stat = 0x3 = Add
stores valet password hash, associates it with key index 63
LBIAlert_Rsp with Event = 0x04, Source = 0x00, Bytes 6-11 = [timestamp], Byte 12 = 0x01, Byte 13 = [Key ID]
BackupIgnition_Rsp with RspCode = 0x0A, RspStatus = 0x0B, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = Valet Password, KeyIndex = 0x00, PhoneName = EOS
displays message that says Valet Mode is active, shows valet password, and explains how to use password

### 2.1.6 Starting vehicle with valet password

#### *2.1.6.1 Requirements*

**[LBI.R103.05]** When SYNC receives *IgnPsswrdDsply_B_Rq = 0x1 = Active* and *Ignition_Status = 0x1 = Off* and the status of Enhanced Valet Mode in SYNC is active, SYNC shall enter Infotainment Mode and display either:

1. The valet password entry screen if *IgnPsswrdLckout_B_Stat = 0x0 = Inactive and Vehicle Connectivity is enabled* OR
2. The lockout popup if *IgnPsswrdLckout_B_Stat = 0x1 = Active and Vehicle Connectivity is enabled*

**[LBI.R347.01]** When the user enters a password at the valet password entry screen, SYNC shall request a challenge from the BLEM (*BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request, Byte 5 = 0x00*).

**[LBI.R348.01]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request*, it shall issue a challenge to SYNC with cryptographic nonce and salt (*BackupIgnition_Rsp* with *RspCode = 0x01 = Issue Challenge, RspStatus = 0x00 = Reserved, Byte 6 = 0x00, Challenge Nonce = Challenge Nonce, Salt = Salt, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R349.01]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request*, it shall compute, using the cryptographic nonce, another hash of all stored password hashes.

**[LBI.R350.02]** SYNC shall compute a hash of entered password using received salt and then compute a hash of this result using received nonce.

**[LBI.R351.01]** SYNC shall respond to the challenge from the BLEM (*BackupIgnition_Rsp* with *RspCode = 0x01 = Issue Challenge*) with computed password hash (*BackupIgnition_Rq* with *OpCode = 0x0F = Valet Start Challenge Response, KeyIndex = EOS, Password = Challenge Password, KeypadCode = EOS*).

**[LBI.R251.03]** When the status of Enhanced Valet Mode in the BLEM is active and the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x0F = Valet Start Challenge Response, KeyIndex = EOS, Password = Challenge Password, KeypadCode = EOS*, the BLEM shall only check the received password hash against the valet password hash.

*Note: This means that, when attempting to start the vehicle while in Enhanced Valet mode, only valid valet passwords will be accepted. Valid backup passwords will not be accepted. To exit Enhanced Valet Mode, the user must start the vehicle via valet password, key fob or PaaK device and then select to exit from the SYNC HMI.*

**[LBI.R352.01]** When the BLEM receives a challenge hash from SYNC (*BackupIgnition_Rq* with *OpCode = 0x02 = Challenge Response, KeyIndex = EOS, Password = Challenge Password, KeypadCode = EOS*), it shall compare it with the hashes that it computed for the stored passwords.

**[LBI.R353.01]** If the BLEM determines that the received password is valid i.e. challenge hash matches a calculated password hash, it shall notify SYNC of this (*BackupIgnition_Rsp with RspCode = 0x02 = Challenge Response Acknowledge, RspStatus = 0x0F = Valid Password, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*) and start a 21-second authorization period.

**[LBI.R354.01]** When the BLEM is in a 21-second authorization period while *Ignition_Status = 0x1 = Off*, the BLEM shall respond positively (*PaakCtlType_D_Stat = 0x1 = Valid, PaakCtlIndx1_No_Actl = [Index]*) to BCM Crypto Start searches (*PaakTrgtType_D_Rq = 0x1 = Crypto, PaakTrgtZone_D_Rq = 0x1 = Interior)*, but negatively to Registry or Polling searches. After this period expires, the BLEM shall respond negatively (*PaakCtlType_D_Stat = 0x1 = Invalid*) to BCM Crypto Start searches.

**[LBI.R252.04]** When the status of Enhanced Valet Mode in the BLEM is active and the BLEM responds positively to a Crypto Start search during 21-second authorization period, it shall report this to the TCU (*LBIAlert_St* with *Event = 0x05 = Valet Password Used, Source = 0x00, [timestamp], [Key ID]*) in a BLEM SyncP signed packet (Service Type 0x40/Sub-Service 0x0) as defined in Transfer Protocol BLEM SPSS.

***Note:*** *Reference TP BLEM SPSS and LBI SPSS for BLEM SyncPPacket definition and its payload. Key ID here refers to the Key ID of the PaaK device associated with the entered backup password.*

**[LBI.R315.02]** If the BLEM starts a lockout timer and it is in Enhanced Valet Mode, it shall report this to the TCU (*LBIAlert_St* with *Event = 0x08 = Lockout in Valet Mode, Source = 0x00, [timestamp], [Key ID]*) in a BLEM SyncP signed packet (Service Type 0x40/Sub-Service 0x1) as defined in Transfer Protocol BLEM SPSS.

***Note:*** *Reference TP BLEM SPSS and LBI SPSS for BLEM SyncPPacket definition and its payload. Key ID here refers to either:*

- *The Key IDs of the PaaK devices that were in the vehicle at the time of valet password generation, if Enhanced Valet Mode was authorized by a PaaK device* OR
- *The Key ID of the PaaK device associated with the entered backup password, if Enhanced Valet mode was authorized by a backup password*

#### *2.1.6.2 Use Case*

| **Actors** | User |
|---|---|
| **Pre-conditions** | User has previously created a valet password.<br>Vehicle is in Enhanced Valet Mode.<br>Vehicle is locked.<br>User is outside vehicle.<br>No associated key fobs or phones-as-keys are near the vehicle. |
| **Scenario Description** | 1. User approaches vehicle.<br>2. User enters valid keypad code.<br>3. Vehicle unlocks.<br>4. User opens door and enters vehicle.<br>5. User presses brake pedal.<br>6. SYNC displays password entry screen.<br>7. Without being inactive for more than 30 seconds, user enters valid valet password via SYNC. (This includes inputting the password then selecting Enter.)<br>8. SYNC displays message instructing user to start the vehicle.<br>9. Within 20 seconds, user presses start button while holding brake pedal.<br>10. Vehicle starts with engine running. |
| **Post-conditions** | User is able to drive away vehicle.<br>Notification that PaaK Backup has been used is sent to user. |
| **List of Exceptions** | User does not enter valid keypad code.<br>User does not enter valid password.<br>User is inactive for more than 30 seconds while SYNC displays password entry screen.<br>User does not start vehicle within 20 seconds of successful password entry.<br>User presses start button without holding brake pedal after password is accepted. |
| **Interfaces** | APIM<br>BCM<br>BLEM<br>TCU<br>SDN<br>PaaK FI |

#### *2.1.6.3 Sequence Diagram*

See *2.1.2.3.*

### 2.1.7 Deleting valet password and keypad code

#### *2.1.7.1 Requirements*

**[LBI.R104.02]** When the user selects the option to exit Valet Mode within SYNC HMI, SYNC shall query the BLEM for PaaK devices in the vehicle (*BackupIgnition_Rq* with *OpCode = 0x06 = Check for Keys to Exit Valet Mode, Byte 5 = 0x00*).

**[LBI.R253.02]** The BLEM shall trigger a BCM Interior Registry search (*FobTrgtType_D_Rq = 0x2 = Registry, FobTrgtZone_D_Rq = 0x1 = Interior, FobTrgtPssvData_No_Rq, FobTrgtRollCode_No_Rq*) whenever it receives *BackupIgnition_Rq* with *OpCode = 0x06 = Check for Keys to Exit Valet Mode*.

***Note: See details of BCM Interior Registry search in Section 3.1.4***

**[LBI.R254.02]** After completing BLEM-requested Interior Registry search (*FobTrgtType_D_Rq = 0x2 = Registry, FobTrgtZone_D_Rq = 0x1 = Interior, FobTrgtPssvData_No_Rq, FobTrgtRollCode_No_Rq* the BCM shall report to the BLEM whether any key fobs were detected in the vehicle (*FobCtlType_D_Stat*, *FobCtlPssvData_No_Actl, FobCtlRollCode_No_Stat*).

**[LBI.R105.03]** After receiving Interior Registry search results from the BCM, the BLEM shall either:

1. Delete the valet password hash and any key indexes that were stored at the time of Enhanced Valet Mode activation (**LBI.R238.03**], if either a PaaK device or a key fob is detected inside the vehicle (to start Exiting Enhanced Valet Mode Option 1) OR
2. Notify SYNC, if neither a PaaK device nor a key fob are detected inside the vehicle (to start Exiting Enhanced Valet Option 2).
    - *BackupIgnition_Rsp* with *RspCode = 0x06 = Check for Keys to Exit Valet Mode, RspStatus = 0x1C = No PaaK and No Fob In Vehicle, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*

*Exiting Enhanced Valet Mode Option 1 starts here.*

**[LBI.R355.01]** When the BLEM cannot delete the valet password hash, it shall notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x06 = Check for Keys to Exit Valet Mode, RspStatus = 0x18 = Password Deleted Failed, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R356.01]** When SYNC receives deletion failure response (*BackupIgnition_Rsp* with *RspCode = 0x06 = Check for Keys to Exit Valet Mode, RspStatus = 0x18 = Password Deleted Failed*), the SYNC HMI shall notify user of unsuccessful valet password deletion/exit of Enhanced Valet Mode.

**[LBI.R255.02]** When the BLEM deletes a valet password hash, it shall report this to the TCU (*LBIAlert_St* with *Event = 0x06 = Valet Password Deleted, Source = 0x02, 0x03, or 0x04, [timestamp], [Key ID]*) in a BLEM SyncP signed packet (Service Type 0x40/Sub-Service 0x0) as defined in Transfer Protocol BLEM SPSS.

***Note:*** *Reference TP BLEM SPSS and LBI SPSS for BLEM SyncPPacket definition and its payload. Key ID here refers to either:*

- *The Key IDs of the PaaK devices that were in the vehicle at the time of valet password generation, if Enhanced Valet Mode was authorized by a PaaK device* OR
- *The Key ID of the PaaK device associated with the entered backup password, if Enhanced Valet mode was authorized by a backup password*

**[LBI.R256.02]** When the BLEM deletes a valet password hash, it shall initiate valet keypad code deletion by responding to periodic RKE challenge from the BCM (*PaaKTrgtActvData_No_Rq*) and sending a request to the BCM to enter keypad programming mode (*KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode*).

***Note: See details of BLEM-BCM Keypad programming requirements in Section 3.1.5***

**[LBI.R106.03]** When the BCM receives *KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode* together with valid RKE response data from the BLEM, it shall enter keypad programming mode and notify the BLEM (*KeyPadCodeProg_D_Stat = 0x2 = LearningMode*) within two seconds and continue to provide notification for up to two seconds.

**[LBI.R107.02]** When the BLEM receives *KeyPadCodeProg_D_Stat = 0x2 = LearningMode*, it shall send to the BCM the valet key index (*PaaKCtlActv_No_Actl = 63*) and a request to delete the valet keypad code (*KeyPadCodeProg_D_Rq = 0x3 = Delete*).

**[LBI.R108.02]** When the BCM receives the request to delete the valet keypad code (*KeyPadCodeProg_D_Rq = 0x3 = Delete*) together with the valet key index (*PaaKCtlActv_No_Actl = 63*), it shall delete the valet keypad code.

**[LBI.R257.01]** When the BCM deletes the valet keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x4 = Delete*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

**[LBI.R316.01]** When the BCM cannot delete the valet keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x5 = ProgrammingFailture*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

**[LBI.R317.02]** When the BLEM receives programming failure response (*KeyPadCodeProg_D_Stat = 0x5 = ProgrammingFailure*), it shall notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x06 = Check for Keys to Exit Valet Mode, RspStatus = 0x1E = Password Deleted Successfully, but Keypad Code Deleted Failed, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R318.02]** When SYNC receives programming failure response (*BackupIgnition_Rsp* with *RspCode = 0x06 = Check for Keys to Exit Valet Mode, RspStatus = 0x1E = Password Deleted Successfully, but Keypad Code Deleted Failed*), the SYNC HMI shall notify user of successful valet password deletion/unsuccessful valet keypad code deletion and then exit Enhanced Valet Mode.

**[LBI.R258.02]** When the BLEM receives confirmation of keypad code deletion (*KeyPadCodeProg_D_Stat = 0x4 = Delete*), it shall notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x06 = Check for Keys to Exit Valet Mode, RspStatus = 0x17 = Password Deleted Successfully, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R109.03]** When SYNC receives confirmation of keypad code deletion (*BackupIgnition_Rsp* with *RspCode = 0x06 = Check for Keys to Exit Valet Mode, RspStatus = 0x17 = Password Deleted Successfully*), the SYNC HMI shall exit Enhanced Valet mode.

*Exiting Enhanced Valet Mode Option 2 starts here.*

**[LBI.R259.02]** When SYNC receives *BackupIgnition_Rsp* with *RspCode = 0x06 = Check for Keys to Exit Valet Mode, RspStatus = 0x1C = No PaaK and No Fob In Vehicle*, the SYNC HMI shall display the backup password entry screen and ask the user to enter a backup password.

**[LBI.R260.02]** When the user enters a password at the backup password entry screen, SYNC shall request a challenge from the BLEM (*BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request, Byte 5 = 0x00*).

**[LBI.R261.01]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request*, it shall issue a challenge to SYNC with cryptographic nonce and salt (*BackupIgnition_Rsp* with *RspCode = 0x01 = Issue Challenge, RspStatus = 0x00 = Reserved, Byte 6 = 0x00, Challenge Nonce = Challenge Nonce, Salt = Salt, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R262.01]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request*, it shall compute, using the cryptographic nonce, another hash of all stored password hashes.

**[LBI.R263.01]** SYNC shall compute a hash of entered password using received salt and then compute a hash of this result using received nonce.

**[LBI.R264.01]** SYNC shall respond to the challenge from the BLEM (*BackupIgnition_Rsp* with *RspCode = 0x01 = Issue Challenge*) with computed password hash (*BackupIgnition_Rq* with *OpCode = 0x0B = Valet Delete Challenge Response, KeyIndex = 0x00, Password = Challenge Password, KeypadCode = EOS*).

**[LBI.R265.01]** When the BLEM receives a challenge hash from SYNC (*BackupIgnition_Rq* with *OpCode = 0x0B = Valet Delete Challenge Response, KeyIndex = 0x00, Password = Challenge Password, KeypadCode = EOS*), it shall compare it with the hashes that it computed for the stored passwords.

**[LBI.R266.02]** If the BLEM determines that the received password is invalid i.e. challenge hash does not match a calculated password hash, it shall increment invalid password counter and then notify SYNC (*BackupIgnition_Rsp with RspCode = 0x0B = Valet Delete Challenge Response Acknowledge, RspStatus = 0x10 = Invalid Password, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R267.02]** When SYNC receives an invalid password notification (*BackupIgnition_Rsp with RspCode = 0x0B = Valet Delete Challenge Response Acknowledge, RspStatus = 0x10 = Invalid Password*), the SYNC HMI shall notify the user that the entered password is invalid and provide an option to retry.

**[LBI.R268.01]** If the BLEM determines that the received password is valid i.e. challenge hash matches a calculated password hash, it shall delete the valet password.

**[LBI.R357.01]** When the BLEM cannot delete the valet password hash, it shall notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x06 = Check for Keys to Exit Valet Mode, RspStatus = 0x18 = Password Deleted Failed, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R358.01]** When SYNC receives deletion failure response (*BackupIgnition_Rsp* with *RspCode = 0x06 = Check for Keys to Exit Valet Mode, RspStatus = 0x18 = Password Deleted Failed*), the SYNC HMI shall notify user of unsuccessful valet password deletion/exit of Enhanced Valet Mode.

**[LBI.R269.03]** When the BLEM deletes a valet password hash, it shall report this to the TCU (*LBIAlert_St* with *Event = 0x06 = Valet Password Deleted, Source = 0x01 = Password, [timestamp], [Key ID]*) in a BLEM SyncP signed packet (Service Type 0x40/Sub-Service 0x0) as defined in Transfer Protocol BLEM SPSS.

***Note:*** *Reference TP BLEM SPSS and LBI SPSS for BLEM SyncPPacket definition and its payload. Key ID here refers to either:*

- *The Key IDs of the PaaK devices that were in the vehicle at the time of valet password generation, if Enhanced Valet Mode was authorized by a PaaK device* OR
- *The Key ID of the PaaK device associated with the entered backup password, if Enhanced Valet mode was authorized by a backup password*

**[LBI.R270.02]** After the BLEM deletes a valet password hash, it shall initiate valet keypad code deletion by responding to periodic RKE challenge from the BCM (*PaaKTrgtActvData_No_Rq*) and sending a request to the BCM to enter keypad programming mode (*KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode*).

***Note: See details of BLEM-BCM Keypad programming requirements in Section 3.1.5***

**[LBI.R271.02]** When the BCM receives *KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode* together with valid RKE response data from the BLEM, it shall enter keypad programming mode and notify the BLEM (*KeyPadCodeProg_D_Stat = 0x2 = LearningMode*) within two seconds and continue to provide notification for up to two seconds.

**[LBI.R272.01]** When the BLEM receives *KeyPadCodeProg_D_Stat = 0x2 = LearningMode*, it shall send to the BCM the valet key index (*PaaKCtlActv_No_Actl = 63*) and a request to delete the valet keypad code (*KeyPadCodeProg_D_Rq = 0x3 = Delete*).

**[LBI.R273.01]** When the BCM receives the request to delete the valet keypad code (*KeyPadCodeProg_D_Rq = 0x3 = Delete*) together with the valet key index (*PaaKCtlActv_No_Actl = 63*), it shall delete the valet keypad code.

**[LBI.R274.01]** When the BCM deletes the valet keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x4 = Delete*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

**[LBI.R319.01]** When the BCM cannot delete the valet keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x5 = ProgrammingFailture*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

**[LBI.R320.02]** When the BLEM receives programming failure response (*KeyPadCodeProg_D_Stat = 0x5 = ProgrammingFailure*), it shall notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x06 = Check for Keys to Exit Valet Mode, RspStatus = 0x1E = Password Deleted Successfully, but Keypad Code Deleted Failed, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R321.02]** When SYNC receives programming failure response (*BackupIgnition_Rsp* with *RspCode = 0x06 = Check for Keys to Exit Valet Mode, RspStatus = 0x1E = Password Deleted Successfully, but Keypad Code Deleted Failed*), the SYNC HMI shall notify user of successful valet password deletion/unsuccessful valet keypad code deletion and then exit Enhanced Valet Mode.

**[LBI.R275.02]** When the BLEM receives confirmation of keypad code deletion (*KeyPadCodeProg_D_Stat = 0x4 = Delete*), it shall notify SYNC (*BackupIgnition_Rsp* with *RspCode = 0x0B = Valet Delete Response Acknowledge, RspStatus = 0x17 = Password Deleted Successfully, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R276.03]** When SYNC receives confirmation of keypad code deletion (*BackupIgnition_Rsp* with *RspCode = 0x0B = Valet Delete Response Acknowledge, RspStatus = 0x17 = Password Deleted Successfully*), the SYNC HMI shall exit Enhanced Valet mode.

#### *2.1.7.2 Use Case*

| **Actors** | User |
|---|---|
| **Pre-conditions** | User has previously created a backup password.<br>Vehicle is in RUN.<br>Vehicle is in Enhanced Valet Mode. |

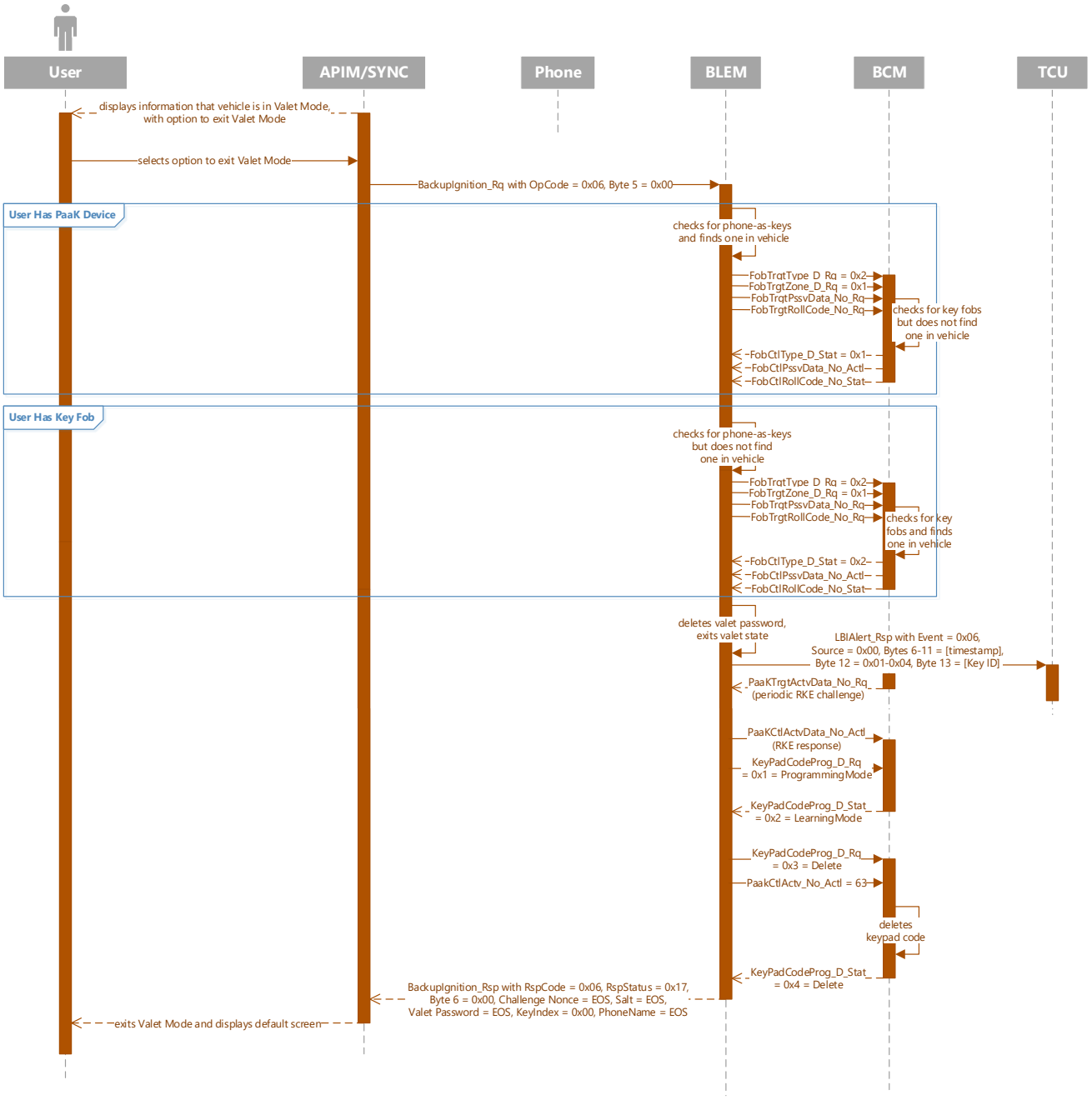
| | |
|---|---|
| | User is inside vehicle.<br>One associated PaaK device is inside the vehicle. |
| **Scenario Description** | 1. User presses button to Exit Valet Mode.<br>2. SYNC displays default screen. |
| **Post-conditions** | Vehicle is no longer in Enhanced Valet Mode.<br>Notification that valet password has been deleted is sent to user. |
| **List of Exception Use Cases** | |
| **Interfaces** | APIM<br>BCM<br>BLEM<br>TCU<br>SDN<br>PaaK FI |

#### *2.1.7.3 Sequence Diagram*

##### 2.1.7.3.1 Deleting valet password and keypad code – phone and/or key fob present

Pre-conditions

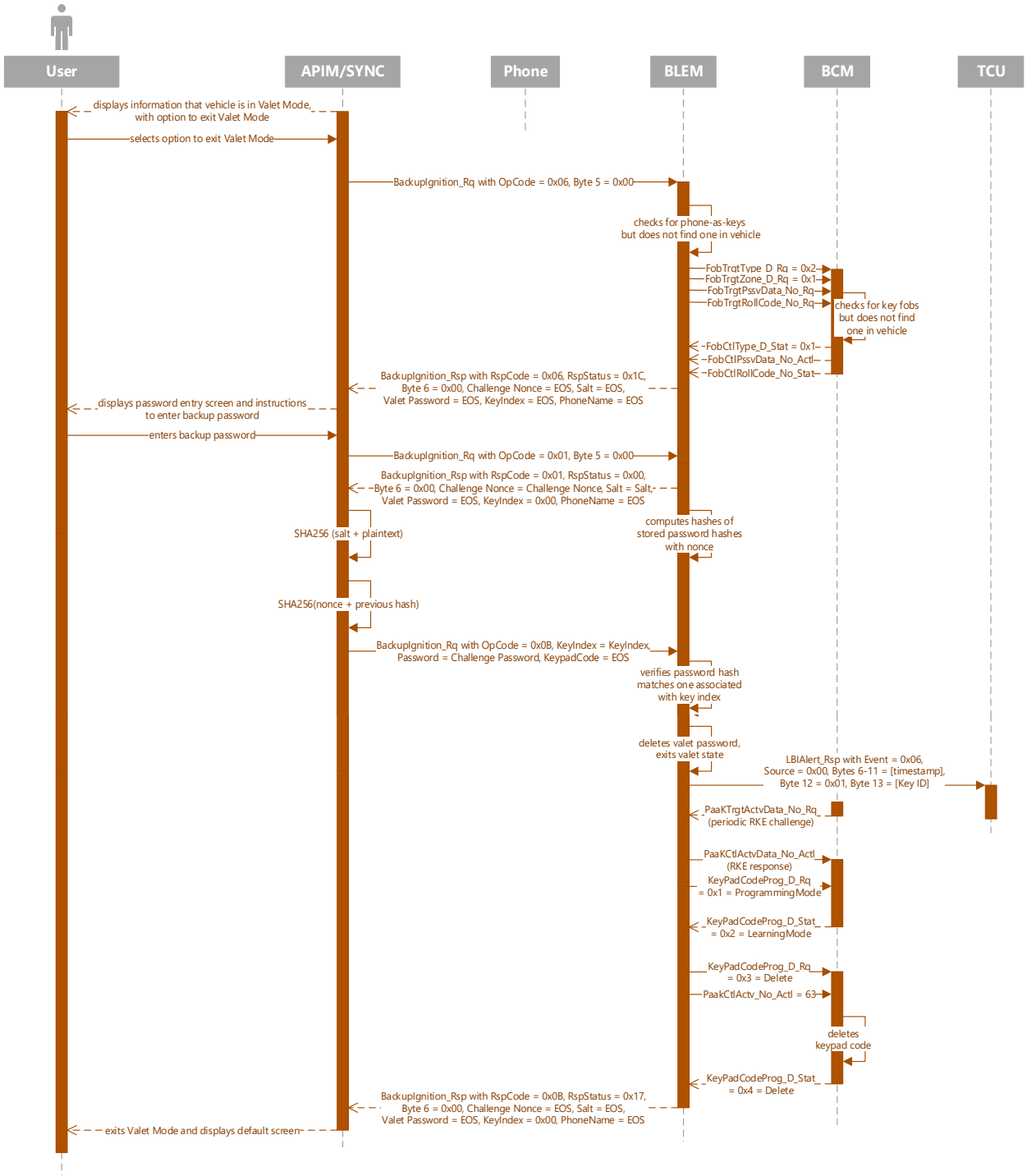
1. User has previously created a valet password.
2. Vehicle is in RUN and user is inside vehicle.
3. Vehicle is in Enhanced Valet Mode.




### 2.1.7.3.2 Deleting valet password and keypad code – no devices present

Pre-conditions

1. User has previously created a valet password.
2. Vehicle is in RUN and user is inside vehicle.
3. Vehicle is in Enhanced Valet Mode.


User
APIM/SYNC
Phone
BLEM
BCM
TCU
displays information that vehicle is in Valet Mode, with option to exit Valet Mode
selects option to exit Valet Mode
BackupIgnition_Rq with OpCode = 0x06, Byte 5 = 0x00
checks for phone-as-keys but does not find one in vehicle
FobTrgtType_D_Rq = 0x2
FobTrgtZone_D_Rq = 0x1
FobTrgtPssvData_No_Rq
FobTrgtRollCode_No_Rq
checks for key fobs but does not find one in vehicle
FobCtlType_D_Stat = 0x1
FobCtlPssvData_No_Actl
FobCtlRollCode_No_Stat
BackupIgnition_Rsp with RspCode = 0x06, RspStatus = 0x1C, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = EOS, PhoneName = EOS
displays password entry screen and instructions to enter backup password
enters backup password
BackupIgnition_Rq with OpCode = 0x01, Byte 5 = 0x00
BackupIgnition_Rsp with RspCode = 0x01, RspStatus = 0x00, Byte 6 = 0x00, Challenge Nonce = Challenge Nonce, Salt = Salt, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
computes hashes of stored password hashes with nonce
SHA256 (salt + plaintext)
SHA256(nonce + previous hash)
BackupIgnition_Rq with OpCode = 0x0B, KeyIndex = KeyIndex, Password = Challenge Password, KeypadCode = EOS
verifies password hash matches one associated with key index
deletes valet password, exits valet state
LBIAlert_Rsp with Event = 0x06, Source = 0x00, Bytes 6-11 = [timestamp], Byte 12 = 0x01, Byte 13 = [Key ID]
PaaKTrgtActvData_No_Rq (periodic RKE challenge)
PaaKCtlActvData_No_Actl (RKE response)
KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode
KeyPadCodeProg_D_Stat = 0x2 = LearningMode
KeyPadCodeProg_D_Rq = 0x3 = Delete
PaakCtlActv_No_Actl = 63
deletes keypad code
KeyPadCodeProg_D_Stat = 0x4 = Delete
BackupIgnition_Rsp with RspCode = 0x0B, RspStatus = 0x17, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS
exits Valet Mode and displays default screen

## 2.2 Secondary Functions

### 2.2.1 Deleting all backup passwords via Master or PaaK Reset

#### *2.2.1.1 Requirements*

**[LBI.R123.01]** When the user selects Master or PaaK Reset through SYNC Settings, the SYNC HMI shall inform the user that all PaaK Backup passwords and associated keypad codes will be erased.

**[LBI.R277.02]** Once the user follows through with a Master or PaaK Reset, the BLEM shall delete all backup password hashes.

**[LBI.R380.01]** When the user follows through with a Master or PaaK Reset and the BLEM is unable to remove backup password hashes from memory, it shall disable backup password feature functionality.

**[LBI.R278.02]** When deleting all backup password hashes, the BLEM shall determine whether there are any keypad codes associated with these password hashes.

If there are any associated keypad codes, the BLEM shall initiate deletion of all PaaK-associated keypad codes by responding (with *PaaKCtlActvData_No_Actl*) to periodic RKE challenge from the BCM (*PaaKTrgtActvData_No_Rq*) and sending a request to the BCM to enter keypad programming mode (*KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode*).

***Note: See details of BLEM-BCM Keypad programming requirements in Section 3.1.5***

**[LBI.R279.02]** When the BCM receives *KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode* together with valid RKE response data from the BLEM, it shall enter keypad programming mode and notify the BLEM (*KeyPadCodeProg_D_Stat = 0x2 = LearningMode*) within two seconds and continue to provide notification for up to two seconds.

**[LBI.R280.01]** When the BLEM receives *KeyPadCodeProg_D_Stat = 0x2 = LearningMode*, it shall send to the BCM a request to delete all personal keypad codes (*KeyPadCodeProg_D_Rq = 0x4 = DeleteAll*).

**[LBI.R124.02]** When the BCM receives the request to delete all personal keypad codes (*KeyPadCodeProg_D_Rq = 0x4 = DeleteAll*), it shall delete all personal keypad codes.

**[LBI.R281.01]** When the BCM deletes all personal keypad codes, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x4 = DeleteAll*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

### 2.2.2 Deleting backup password via key revoke

#### *2.2.2.1 Requirements*

**[LBI.R125.02]** When the user revokes the CAK for their PaaK device, the mobile app HMI shall inform the user that, if they have created a backup password and keypad code for their device, these will be deleted at the completion of the revoke request.

**[LBI.R126.03]** When the BLEM receives a request to revoke the CAK for a specific PaaK device, it shall delete the backup password hash associated with that device.

**[LBI.R282.02]** When deleting a backup password hash as a result of a CAK revoke, the BLEM shall determine whether there is a keypad code associated with that CAK.

If there is an associated keypad code, the BLEM shall initiate keypad code deletion by responding (with *PaaKCtlActvData_No_Actl*) to periodic RKE challenge from the BCM (*PaaKTrgtActvData_No_Rq*) and sending a request to the BCM to enter keypad programming mode (*KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode*).

***Note: See details of BLEM-BCM Keypad programming requirements in Section 3.1.5***

**[LBI.R359.01]** When the BCM receives *KeyPadCodeProg_D_Rq = 0x1 = ProgrammingMode* together with valid RKE response data from the BLEM, it shall enter keypad programming mode and notify the BLEM (*KeyPadCodeProg_D_Stat = 0x2 = LearningMode*) within two seconds and continue to provide notification for up to two seconds.

**[LBI.R360.01]** When the BLEM receives *KeyPadCodeProg_D_Stat = 0x2 = LearningMode*, it shall send to the BCM the key index associated with the CAK (*PaaKCtlActv_No_Actl = [index]*) and a request to delete the personal keypad code (*KeyPadCodeProg_D_Rq = 0x3 = Delete*).

**[LBI.R361.01]** When the BCM receives the request to delete the personal keypad code (*KeyPadCodeProg_D_Rq = 0x3 = Delete*) together with the key index (*PaaKCtlActv_No_Actl = [index]*), it shall delete the personal keypad code that is associated with the received key index.

**[LBI.R362.01]** When the BCM deletes a keypad code, it shall notify the BLEM (*KeyPadCodeProg_D_Stat = 0x4 = Delete*) for one second and then exit programming mode (*KeyPadCodeProg_D_Stat = 0x0 = NormalMode*).

### 2.2.3 Transitioning vehicle from non-motive to motive state with backup password

#### *2.2.3.1 Requirements*

**[LBI.R128.01]** User shall be able to transition vehicle from non-motive to motive state using PaaK Backup.

#### *2.2.3.2 Use Case*

| | |
|---|---|
| **Actors** | User |
| **Pre-conditions** | User has previously activated Phone-as-a-Key for their vehicle via Lincoln mobile app.<br>User is logged into Lincoln app on their mobile phone.<br>User's mobile phone and vehicle are BT connected.<br>User has previously created backup password.<br>Vehicle is locked. |
| **Scenario Description** | 1. User remote starts vehicle via mobile app.<br>2. Phone becomes disabled (e.g. battery drained)<br>3. User approaches locked vehicle, cannot enter.<br>4. User enters valid keypad code.<br>5. Vehicle unlocks.<br>6. User opens door and enters vehicle.<br>7. User presses brake pedal.<br>8. SYNC displays backup password entry screen.<br>9. Without being inactive for more than 30 seconds, user enters valid backup password via SYNC. (This includes inputting the password then selecting Enter. Touch events on screen extend inactivity timeout.)<br>10. SYNC displays "Password accepted. Start vehicle within 20 seconds. "<br>11. Within 20 seconds, user presses start button while holding brake pedal.<br>12. Vehicle transitions from non-motive to motive state. |
| **Post-conditions** | User is able to drive away vehicle.<br>User is able to charge their PaaK device in vehicle. |
| **List of Exception Use Cases** | User does not enter valid password.<br>User is inactive for more than 30 seconds while SYNC displays password entry screen.<br>User does not start vehicle within 20 seconds of successful password entry. |
| **Interfaces** | BLEM<br>APIM<br>BCM |

### 2.2.4 Exiting secure idle state with backup password

#### *2.2.4.1 Requirements*

**[LBI.R129.04]** When the BCM is in a secure idle state and the conditions below are true, it shall trigger the backup password entry screen (*IgnPsswrdDsply_B_Rq = 0x1 = Active*):

1. Brake pedal is pressed OR
2. Accelerator pedal is pressed OR
3. Shifter button is pressed AND
4. No key fobs or PaaK devices are detected inside the vehicle AND
5. There is a least one backup password created (*IgnPsswrdActv_B_Stat = 0x1 = Active*)

**[LBI.R322.03]** When SYNC receives *IgnPsswrdDsply_B_Rq = 0x1 = Active* and engine status, *PwPckTq_D_Stat = 0x1 - PwPckOn_TqNotAvailable or 0x0 - PwPckOff_TqNotAvailable* and the status of Enhanced Valet Mode in SYNC is inactive, SYNC shall display either:

1. A backup password entry screen if *IgnPsswrdLckout_B_Stat = 0x0 = Inactive and Vehicle Connectivity is enabled*
2. A lockout popup if *IgnPsswrdLckout_B_Stat = 0x1 = Active and Vehicle Connectivity is enabled*

Note:
*PwPckTq_D_Stat = 0x0 - PwPckOff_TqNotAvailable* means engine is not running
*PwPckTq_D_Stat = 0x1 - PwPckOn_TqNotAvailable* means engine is running in NonMotive mode
*PwPckTq_D_Stat = 0x2 – StartInprgrss_TqNotAvail* means engine is cranking
*PwPckTq_D_Stat = 0x3 - PwPckOn_TqAvailable* means engine is running in Motive mode

**[LBI.R323.02]** When the password entry screen is active and *Ignition_Status = 0x4 = Run*, the SYNC HMI shall return to the previous screen after 30 seconds of inactivity. Keyboard button presses on screen and/or additional receptions of *IgnPsswrdDsply_B_Rq = 0x1 = Active* shall reset inactivity timer.

**[LBI.R363.01]** When the lockout popup expires and *Ignition_Status = 0x1 = Run*, the SYNC HMI shall return to current screen.

**[LBI.R364.01]** When the user enters a password at the password entry screen, SYNC shall request a challenge from the BLEM (*BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request, Byte 5 = 0x00*).

**[LBI.R365.01]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request*, it shall issue a challenge to SYNC with cryptographic nonce and salt (*BackupIgnition_Rsp* with *RspCode = 0x01 = Issue Challenge, RspStatus = 0x00 = Reserved, Byte 6 = 0x00, Challenge Nonce = Challenge Nonce, Salt = Salt, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R366.01]** When the BLEM receives *BackupIgnition_Rq* with *OpCode = 0x01 = Challenge Request*, it shall compute, using the cryptographic nonce, another hash of all stored password hashes.

**[LBI.R367.01]** SYNC shall compute a hash of entered password using received salt and then compute a hash of this result using received nonce.

**[LBI.R368.01]** SYNC shall respond to the challenge from the BLEM (*BackupIgnition_Rsp* with *RspCode = 0x01 = Issue Challenge*) with computed password hash (*BackupIgnition_Rq* with *OpCode = 0x02 = Challenge Response, KeyIndex = EOS, Password = Challenge Password, KeypadCode = EOS*).

**[LBI.R369.01]** When the BLEM receives a challenge hash from SYNC (*BackupIgnition_Rq* with *OpCode = 0x02 = Challenge Response, KeyIndex = EOS, Password = Challenge Password, KeypadCode = EOS*), it shall compare it with the hashes that it computed for the stored passwords.

**[LBI.R370.01]** If the BLEM determines that the received password is invalid i.e. challenge hash does not match a calculated password hash, it shall increment invalid password counter and then notify SYNC (*BackupIgnition_Rsp with RspCode = 0x02 = Challenge Response Acknowledge, RspStatus = 0x10 = Invalid Password, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*).

**[LBI.R371.01]** When SYNC receives an invalid password notification (*BackupIgnition_Rsp with RspCode = 0x02 = Challenge Response Acknowledge, RspStatus = 0x10 = Invalid Password*), the SYNC HMI shall notify the user that the entered password is invalid and provide an option to retry.

**[LBI.R034.03]** If the BLEM determines that the received password is valid i.e. challenge hash matches a calculated password hash, it shall notify SYNC of this (*BackupIgnition_Rsp with RspCode = 0x02 = Challenge Response Acknowledge, RspStatus = 0x0F = Valid Password, Byte 6 = 0x00, Challenge Nonce = EOS, Salt = EOS, Valet Password = EOS, KeyIndex = 0x00, PhoneName = EOS*) and start a 21-second authorization period.

**[LBI.R324.02]** When the BLEM is in a 21-second authorization period while *Ignition_Status = 0x4 = Run*, the BLEM shall respond positively (*PaakCtlType_D_Stat = 0x1 = Valid, PaakCtlIndx1_No_Actl = [Index]*) to BCM Crypto Start searches (*PaakTrgtType_D_Rq = 0x1 = Crypto, PaakTrgtZone_D_Rq = 0x1 = Interior)*, but negatively to Registry or Polling searches. After this period expires, the BLEM shall respond negatively (*PaakCtlType_D_Stat = 0x1 = Invalid*) to BCM Crypto Start searches.

**[LBI.R325.04]** When SYNC receives a valid notification from the BLEM (*BackupIgnition_Rsp with RspCode = 0x02 = Challenge Response Acknowledge, RspStatus = 0x0F = Valid Password*) and engine status, *PwPckTq_D_Stat = 0x1 - PwPckOn_TqNotAvailable*, the SYNC HMI shall notify the user that the entered password has been accepted and that they can now press the brake and shift out of park in order to drive the vehicle.
This message shall display for 20 seconds unless vehicle engine status *PwPckTq_D_Stat* changes to *PwPckOn_TqAvailable* (engine running and in Motive mode) or *PwPckOff_TqNotAvailable* (engine is not running), then this message shall be dismissed.

**[LBI.R383.01]** When SYNC receives a valid notification from the BLEM (*BackupIgnition_Rsp with RspCode = 0x02 = Challenge Response Acknowledge, RspStatus = 0x0F = Valid Password*) and engine status, *PwPckTq_D_Stat = 0x0 - PwPckOff_TqNotAvailable*, the SYNC HMI shall notify the user that the entered password has been accepted and that they can now press the brake and start button in order to start the vehicle.
This message shall display for 20 seconds unless the vehicle engine status *PwPckTq_D_Stat* changes to *PwPckOn_TqAvailable* (engine running in Motive mode) or *PwPckOn_TqNotAvailable* (engine is running in NonMotive mode), then this message shall be dismissed.

**[LBI.R283.04]** When the BCM is in a secure idle state and the conditions below are true, it shall exit the secure idle state:

1. Any door transitions from open to closed OR
2. Brake pedal is pressed OR
3. Accelerator pedal is pressed OR
4. Shifter button is pressed OR
5. Seatbelt becomes buckled AND
6. BLEM responds positively to BCM Crypto Start search

The BCM shall suspend secure idle operation until next ignition cycle when the keypad code associated with index 63 is stored in BCM.

**[LBI.R284.02]** When there exists a keypad code associated with key index 63 in the BCM, secure idle operation shall be suspended until the next key cycle.

*Note: This means that when a user activates Enhanced Valet Mode, secure idle will be disabled, even if the vehicle is currently in a secure idle state. This also means that deactivating Enhanced Valet Mode will re-enable secure idle.*

#### *2.2.4.2 Use Case*

| | |
|---|---|
| **Actors** | User |
| **Pre-conditions** | User has previously created backup password.<br>Vehicle is in RUN.<br>Vehicle transmission is in park.<br>User is outside and away from vehicle.<br>Vehicle is unlocked.<br>No associated key fobs or phones-as-keys are inside vehicle. |
| **Scenario Description** | 1. User returns to vehicle.<br>2. User attempts to start engine/shift vehicle out of park.<br>3. Cluster displays “No Key Detected”.<br>4. SYNC displays backup password entry screen.<br>5. Without being inactive for more than 30 seconds, user enters valid backup password via SYNC. (This includes inputting the password then selecting Enter. Touch events on screen extend inactivity timeout.)<br>6. SYNC displays “Password accepted” and returns to Home screen.<br>7. Cluster no longer displays “No Key Detected”. |
| **Post-conditions** | User is able to drive away vehicle |
| **List of Exception Use Cases** | |
| **Interfaces** | APIM<br>BCM<br>BLEM<br>IPC |

# 3 General Requirements

## 3.1 Functional Requirements

### 3.1.1 BLEM

**[LBI.R326.01]** The BLEM shall send *IgnPsswrdActv_B_Stat = 0x1 = Active* when there is a least one backup password stored in HSM.

**[LBI.R408.01]** BLEM shall monitor the following signals:

*Ignition_Status = 0x4 = Run*
*GearLvlPos_D_Actl = 0x0 = Park*

and compare them against an opcode send out by SYNC when a user initiates the functions listed below:

- Create backup passwords/keypad codes for Paak devices
- Reset (change, not delete) backup passwords/keypad codes for Paak devices
- Delete backup passwords/keypad codes for PaaK devices one password at a time
- Delete all backup passwords/keypad codes for PaaK devices at once
- Activate Enhanced Valet Mode (Creating Enhanced Valet password and entering Enhanced Valet Mode)
- Deactivate Enhanced Valet Mode (Deleting Enhanced Valet password and exiting Enhanced Valet Mode)

***Note:*** *This verification check insures that the BLEM is synchronized with the SYNC and does not get too far ahead if the SYNC sends an opcode at the same time the user turns the ignition. This guarantees that response to the SYNC does not get lost, etc.*

**[LBI.R410.01]** The BLEM shall de-bounce the phone localization state for 2.5 sec per below conditions:

- When the PaaK phone is detected in vehicle interior zone, then this state shall be reflected immediately to LBI function;
- When the PaaK phone is not detected in vehicle interior zone, then de-bounce for 2.5 sec before reporting the status to LBI function.

**[LBI.R285.01]** The BLEM shall stay in TP heartbeat session (see TP BLEM SPSS) no longer than five seconds. If, at the end of five seconds, it has not completed its task, it shall respond with *CES = 0x10 = Final Result – Failure*.

**[LBI.R327.01]** The BLEM shall not assign a CAK to key index 63. This index is reserved for valet key.

**[LBI.R286.02]** If the TCU in the vehicle is not authorized but there are still valid CAKs in the BLEM, the BLEM shall continue to allow verification of backup passwords.

**[LBI.R287.02]** For each key index, the BLEM shall keep track of whether there is:

- an active backup password associated with it.
- an active keypad code associated with it.

**[LBI.R288.03]** When requesting the BCM to store a keypad code, the BLEM shall respond (with *PaaKCtlActvData_No_Actl*) to the periodic RKE challenge from the BCM (*PaaKTrgtActvData_No_Rq*) with keypad code data embedded in the least significant 21 bits.

See below for mapping of button to bit value:

000 = NULL
001 = “1/2” button pressed
010 = “3/4” button pressed
011 = “5/6” button pressed
100 = “7/8” button pressed
101 = “9/0” button pressed

The seven digits of the keypad code are contained in the lowest 21 bits of the 40-bit signal.
Bit 21 is the most significant bit of the keypad code.
Bit 0 is the least significant bit of the keypad code and the 40-bit signal.
Bits 21-39 are not used.

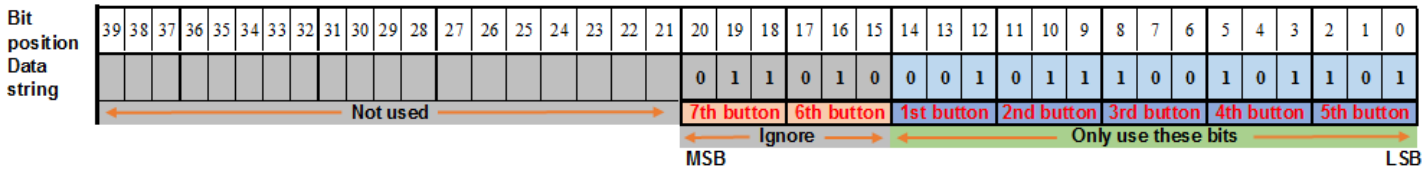
When the BLEM receives TP bit string from the SYNC, it shall verify the vehicle configuration (e.g. there are markets that require the use of 7-digit codes and 5-digit codes), arrange the data into lower 4 bytes of *PaaKCtlActvData_No_Actl* and then transmit it to the BCM.

Bits 0-2 = fifth button pressed.
Bits 3-5 = fourth button pressed.
Bits 6-8 = third button pressed.
Bits 9-11 = second button pressed.
Bits 12-14 = first button pressed.
Bits 15-17 = sixth button pressed.
Bits 18-20 = seventh button pressed.

If the vehicle configuration (as determined by BLEM) does not match the data received from the SYNC (e.g. market with 5-digit code and the 6th & 7th button press parameters are not set to Null), the BLEM shall set *Invalid Data Received From Accessory Protocol Interface Module* DTC, indicating configuration mismatch and transmit the data to the BCM.

***Note:*** *The keypad code is an $N_{th}$-button sequence where each button is represented by three bits. Based on the vehicle configuration, BLEM data shall be represented with 7-digit codes (e.g. 7-button press sequence).*
*When 5-digit codes are implemented, the Sixth and Seventh button press parameters shall be ignored.*

Example of bit string data for keypad code **1579936** as sent out by BLEM and received by BCM:




**[LBI.R289.03]** Upon generating a valet password, the BLEM shall compute its hash and verify its uniqueness by comparing it to existing password hashes. If it is not unique, it shall generate a new 8-digit or 10-digit (based on vehicle configuration) valet password, which it shall also verify for uniqueness.

**[LBI.R290.01]** If the BLEM receives *KeyPadCodeProg_D_Stat = 0x6 = Duplicate* from the BCM when it requests to store a keypad code, the BLEM shall either:

1. Notify SYNC that keypad code is already in use (*BackupIgnition_Rsp* with *RspCode = 0x08 = Keypad Code Create Response, RspStatus = 0x20 Keypad Code Duplicate*), if the request was to store a personal keypad code
2. Generate a new valet password and re-send the first five digits to the BCM, if the request was to store a valet keypad code

**[LBI.R372.01]** As part of the BLEM diagnostic routine used during module swap, the BLEM shall delete all backup passwords and send command to BCM to delete all associated keypad codes. Reference REQ-257978 in BLEM/BLEAM Common Function BLEM SPSS.

### 3.1.2 BCM

**[LBI.R291.01]** When the BCM receives *PaaKCtlIndx1_No_Actl = 63* in response to Crypto Start challenge, it shall start vehicle in MyKey mode.

**[LBI.R292.02]** When the BCM receives a request to store a keypad code (valet or personal) from the BLEM, it shall verify that this code is unique and according to vehicle configuration (e.g. there are markets that require the use of 7-digit codes and 5-digit codes).

If the vehicle configuration (as determined by BCM) does not match the data received from the BLEM (e.g. market with 5-digit codes and the 6th & 7th button press parameters are not set to Null), the BCM shall respond to the BLEM with *KeyPadCodeProg_D_Stat = 0x5 = ProgrammingFailure,* set Invalid Data Received DTC and exit programming mode.

**[LBI.R293.02]** If the BCM determines that a received keypad code is not unique, it shall respond to the BLEM with *KeyPadCodeProg_D_Stat = 0x6 = Duplicate* and increment an attempt counter if the request was to store a valet keypad code. The BCM shall remain in programming mode for up to two seconds.

**[LBI.R294.03]** While in programming mode, the BCM shall accept valet keypad codes re-attempts from the BLEM. After the attempt counter reaches four, the BCM shall accept whatever keypad code it received from the BLEM, even if it is not unique, and then reset attempt counter.

### 3.1.3 SYNC

*Note: In the requirements below "password entry screen" refers to both the backup password entry screen and the valet password entry screen for starting the vehicle.*

**[LBI.R195.02]** If SYNC is running the welcome animation when SYNC receives *IgnPsswrdDsply_B_Rq = 0x1 = Active*, SYNC shall cancel the greeting timer and display the password entry screen after the welcome animation is finished.

**[LBI.R196.03]** After 30 seconds of inactivity at the password entry screen:

- The SYNC HMI shall go to the previous screen if:
  - *Ignition_Status = 0x4 = Run* OR
  - *Ignition_Status = 0x1 = Off* and *Delay_Accy = 0x1 = On* OR
  - *Ignition_Status = 0x1 = Off* and SYNC is in Extended Play mode.
- SYNC shall suspend if:

- *Ignition_Status = 0x1 = Off, Delay_Accy = 0x0 = Off,* and SYNC is not in Extended Play mode.

Keyboard button presses on screen and/or additional receptions of *IgnPsswrdDsply_B_Rq = 0x1 = Active* shall reset inactivity timer.

**[LBI.R306.02]** After the lockout popup expires:

- The SYNC HMI shall go to the previous screen if:
  - *Ignition_Status = 0x4 = Run* OR
  - *Ignition_Status = 0x1 = Off* and *Delay_Accy = 0x1 = On* OR
  - *Ignition_Status = 0x1 = Off* and SYNC is in Extended Play mode.
- SYNC shall suspend if:
  - *Ignition_Status = 0x1 = Off, Delay_Accy = 0x0 = Off, and SYNC is not in Extended Play mode.*

**[LBI.R295.02]** Once SYNC transmits a password hash to the BLEM, it shall delete this password hash from memory.

**[LBI.R296.02]** If SYNC has received multiple challenges nonces, it shall recognize only the most recent one as valid.

**[LBI.R379.02]** When SYNC transmits a keypad code that is associated with a backup password to the BLEM, it shall structure the data as an $N_{th}$ button sequence where each button is represented by three bits.
TP method specification defines the bytes 6-9 of the keypad code when SYNC sends BackupIgnition_Rq with Opcode = 0x08.

The mapping of button to bit value as follows:

000 = NULL
001 = “1/2” button pressed
010 = “3/4” button pressed
011 = “5/6” button pressed
100 = “7/8” button pressed
101 = “9/0” button pressed

*When a customer is creating a keypad code, the SYNC shall provide an appropriate screen (based on vehicle configuration) with either 7-digit or 5-digit codes.*
*Prior to transmitting a keypad code data to the BLEM, the SYNC shall verify the vehicle configuration (e.g. there are markets that require the use of 7-digit codes and 5-digit codes), and then send bit string data.*

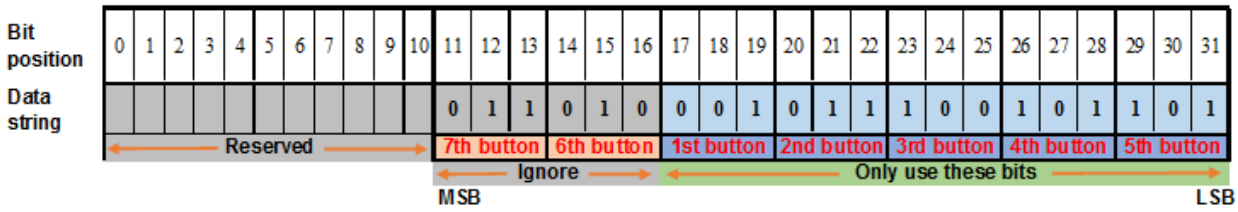
*If the SYNC detects a misconfiguration (for example, if configuration calls for 7-digit codes but the bit string data consists of 5-digit codes), it shall set a Control Module Configuration Incompatible DTC and not transmit a keypad code data to the BLEM.*
*Instead, it shall provide a warning/notification pop up to a customer via HMI interface to indicate an error has occurred and instruct a user to take a vehicle for service.*

***Note:** SYNC data shall always be represented with 7-digit codes (e.g. 7-button press sequence). When 5-digit codes are implemented, the Sixth and Seventh button press parameters shall be set to Null.*

For example, a keypad code of 1234567 consists of keypad buttons **"1/2", "1/2", "3/4", "3/4", "5/6", "5/6", "7/8".**

As a bit string, this is represented as 0000 0000 000 **100**$_{\text{seventh button}}$ **011**$_{\text{sixth button}}$ **001**$_{\text{first button}}$ **001**$_{\text{second button}}$ **010**$_{\text{third button}}$ **010**$_{\text{forth button}}$ **011**$_{\text{fifth button}}$

Here is another example of bit string data for keypad code **1579936** as sent out by SYNC and received by BLEM:




**[LBI.R409.01]** When a customer is creating a personal keypad code, the SYNC shall verify customer acceptable codes based on the vehicle configuration (e.g. there are markets that require the use of 7-digit codes and 5-digit codes).
If SYNC determines the configuration is for 7-digit codes, then it shall apply the following restrictions:

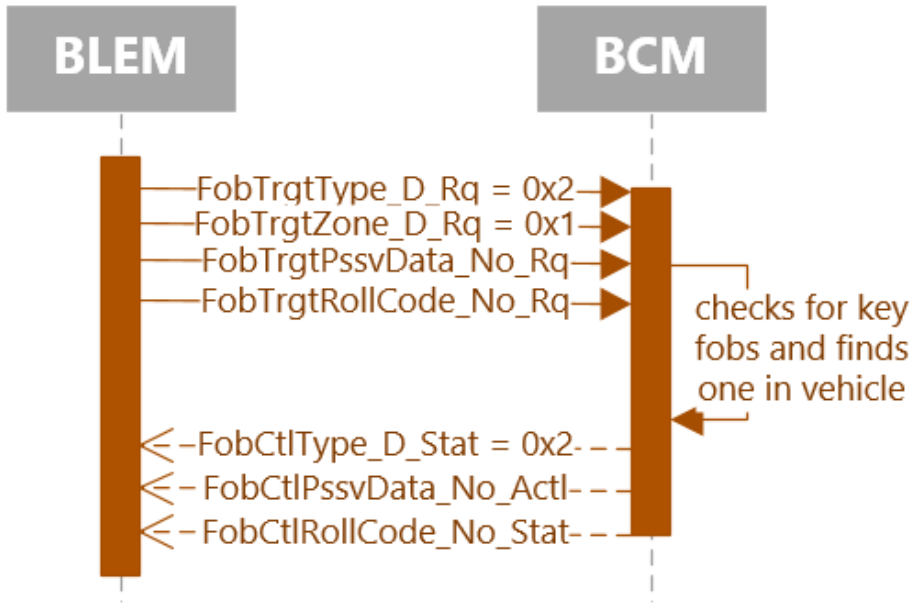
- Desired door keypad code cannot consist of all the same numbers
  - customer presses button **"1/2"** seven (7) times
  - customer presses button **"3/4"** seven (7) times
  - customer presses button **"5/6"** seven (7) times
  - customer presses button **"7/8"** seven (7) times
  - customer presses button **"9/0"** seven (7) times

When a single button has been pressed six (6) consecutive times, that button shall become grey out for the last digit entry and SYNC screen shall also be populated with instruction stating "Desired 7-digit keypad code must not consist of selecting the same button seven times"

### 3.1.4 BLEM-BCM Interface requirements for Interior Registry search (Key Fob search)

**[LBI.R384.01]** The BLEM shall send the following signals to the BCM when requesting a key fob search:

***FobTrgtType_D_Rq*** *(LBIFobTargetSearchType)*
***FobTrgtZone_D_Rq*** *(LBIFobTargetSearchZone)*
***FobTrgtPssvData_No_Rq*** *(LBIFobTargetSearchData)*
***FobTrgtRollCode_No_Rq*** *(LBIFobTargetSearchRollCode)*




*Note:*

***FobTrgtType_D_Rq*** *(LBIFobTargetSearchType)* is a signal indicating the type of search requested by the BLEM.

***FobTrgtZone_D_Rq*** *(LBIFobTargetSearchZone)* is a signal indicating at what zone of the vehicle the BCM should be searching for a key fob.

***FobTrgtPssvData_No_Rq*** *(LBIFobTargetSearchData)* is a signal that includes 5 bytes of the challenge data sent by the BLEM to be used to calculate an authentication response.

***FobTrgtRollCode_No_Rq*** *(LBIFobTargetSearchRollCode)* is a signal that includes rolling count transmitted by the BLEM and used to align a search request with the corresponding search result.

***FobCtlType_D_Stat*** *(LBIFobCtrlSearch_Rslt)* is a signal indicating whether a valid fob was found by BCM during the Interior Registry search.

***FobCtlPssvData_No_Actl*** *(LBIFobCtrlSearchData)* is an array of 5 elements (40 bits) extracted from the encrypted data when the BCM search request for key fob is reached.

***FobCtlRollCode_No_Stat*** *(LBIFobCtrlSearchRollCode)* is a signal that includes rolling count transmitted by the BCM and used to align a search request with the corresponding search result.

From above diagram:

- The search type *LBIFobTargetSearchType* shall be Registry.
- The search zone *LBIFobTargetSearchZone* shall be Interior.
- The search (challenge) data *LBIFobTargetSearchData* shall consist of a 40-bit sequence of random data.
- The search roll code *LBIFobTargetSearchRollCode* shall be equal to the previous roll code value plus one. When the roll code = 15 (0xF), it shall rollover to 0.

**[LBI.R385.001]** When the BLEM requests the BCM to perform a key fob search, it shall also calculate an AES output using AES-128 algorithm. The AES key shall be derived from the BLEM/BCM **TargetID** (**see PaaK SPSS requirement PaaK-REQ-242454**) and the AES input shall be derived from the challenge data *LBIFobTargetSearchData* and roll code *LBIFobTargetSearchRollCode* as seen in **Figure 1** below.

The BLEM shall use the appropriate calculated response, indicated by ***FobCtlType_D_Stat*** *(LBIFobCtrlSearch_Rslt)* and compare it with the challenge response data received via ***FobCtlPssvData_No_Actl*** *(LBIFobCtrlSearchData)*.

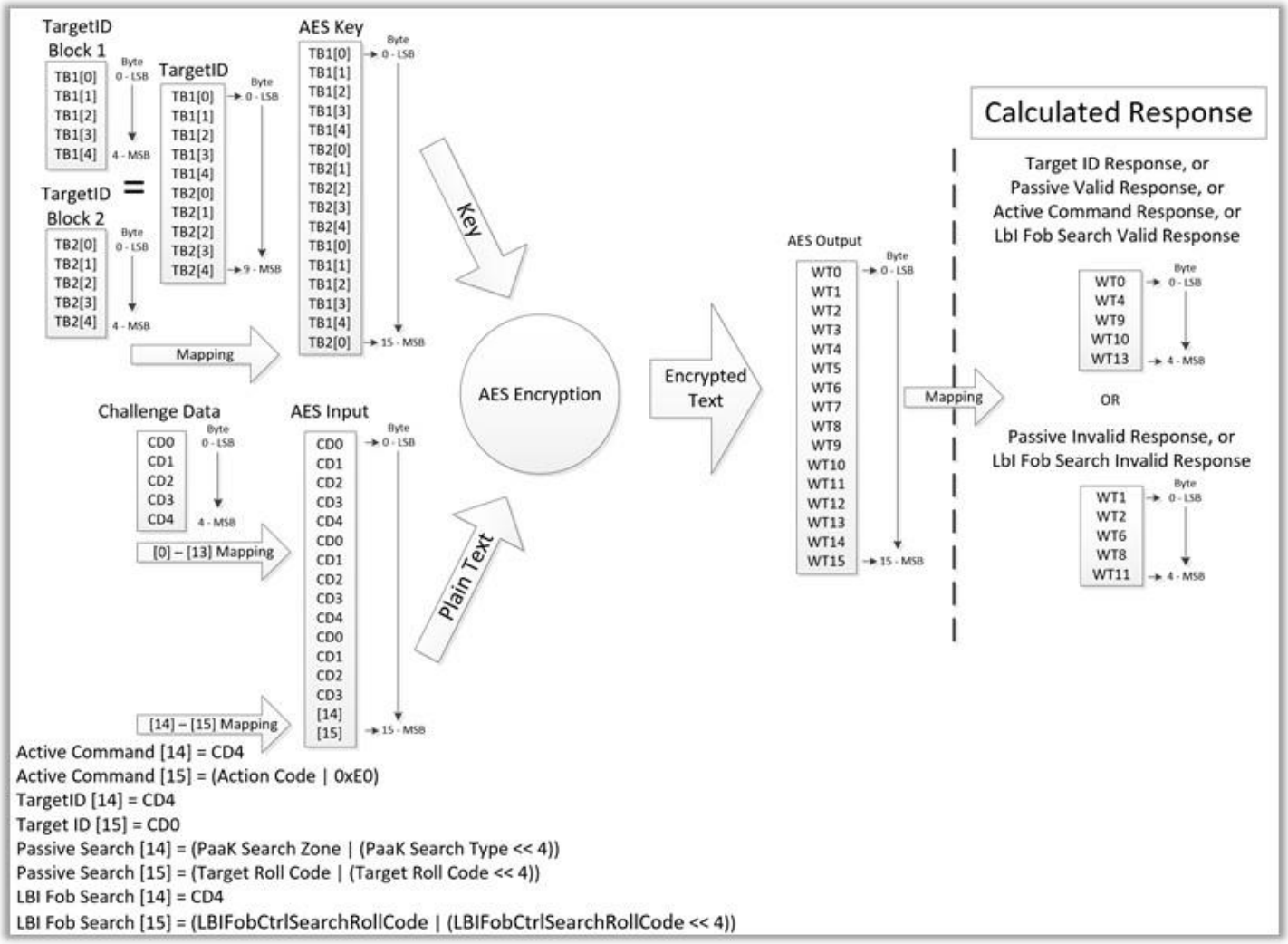



**Figure 1**

***Note:*** *the following definitions are used in above figure 1*

*TB1[0] is TargetID Block 1 byte 0*
*TB2[0] is TargetID Block 2 byte 0*
*TB1[0] & TB2[0] are LSB*
*TB1[4] & TB2[4] are MSB*
*TargetID is comprised of TB1 & TB2, where byte 0 = TB1[0] is LSB and byte 9 = TB2[4] is MSB*

*CD0 is Challenge Data byte 0 = LSB*
*CD4 is Challenge Data byte 4 = MSB*

*AES Key is an encryption key and is comprised of TB1 & TB2 of the TargetID for the first 10 bytes and then TB1 for the next 5 bytes and byte 0 of the TB2 for the byte 15, where byte 0 of the AES Key is LSB and byte 15 is MSB*

*AES Input is a string of data used by AES algorithm and consists of 15 bytes of the Challenge Data with byte 0 through byte 13 plus 2 last bytes that are used for action specific responses. For example, for Active commands we use byte 4 of the Challenge Data for the byte 14 and Action Code for the byte 15 of the AES input. The Action code is the data that is sent in PaakCtlActv_D_Rq.*
*For LBI Key Fob Search function we use byte 4 of the Challenge Data for the byte 14 and FobCtlRollCode_No_Stat for the byte 15 of the AES input.*
*AES input byte 0 is LSB and byte 15 is MSB.*

*AES key and AES input data are used in AES algorithm and will yield 16 byte AES output data, where WT0 is LSB and WT15 is MSB.*
*Depending on specific function, for example when performing a TargetID transfer/exchange or LBI Fob Search, the AES output data will either consist of bytes [0, 4, 9, 10, 13] or bytes [1, 2, 6, 8, 11].*

**[LBI.R386.001] The BCM shall send the following signals to BLEM when responding to a key fob search:**

***FobCtlType_D_Stat*** *(LBIFobCtrlSearch_Rslt)*
***FobCtlPssvData_No_Actl*** *(LBIFobCtrlSearchData)*
***FobCtlRollCode_No_Stat*** *(LBIFobCtrlSearchRollCode)*

Where:

- The search result *LBIFobCtrlSearch_Rslt* shall consist of either a Valid response (key found) or Invalid response (key not found).
- The search (challenge response) data *LBIFobCtrlSearchData* shall consist of a 40-bit sequence of calculated data (see requirement **LBI.R386.001**).
- The search roll code *LBIFobCtrlSearchRollCode* shall match the roll code received from the BLEM.

**[LBI.R387.001]** When the BCM receives the request to perform a key fob search from the BLEM, it shall calculate an AES output using AES-128 algorithm. The AES Key shall be derived from the BLEM/BCM **TargetID** (**see PaaK SPSS requirement PaaK-REQ-242454**) and the AES input shall be derived from the challenge data ***FobTrgtPssvData_No_Rq*** *(LBIFobTargetSearchData)* and the roll code ***FobTrgtRollCode_No_Rq*** *(LBIFobTargetSearchRollCode)* as seen in **Figure 1**.

- The BCM shall echo back the roll code (***FobCtlRollCode_No_Stat = FobTrgtRollCode_No_Rq***) and calculate the Fob Search Challenge Response.
- If a key fob was found (***FobCtlType_D_Stat*** = VALID), the BCM shall populate the challenge response data ***FobCtlPssvData_No_Actl*** *(LBIFobCtrlSearchData)* with bytes [0, 4, 9, 10, 13] of the AES output.

- If a key fob was not found (***FobCtlType_D_Stat*** = INVALID), the BCM shall populate the challenge response data ***FobCtlPssvData_No_Actl*** *(LBIFobCtrlSearchData)* with bytes [1, 2, 6, 8, 11] of the AES output.

**[LBI.R388.001]** When the BLEM receives a key search response from the BCM via the signal ***FobCtlType_D_Stat***, it shall set these signals to null:

***FobTrgtType_D_Rq*** *(*LBIFobTargetSearchType)
***FobTrgtZone_D_Rq*** (LBIFobTargetSearchZone)

**[LBI.R389.001]** When authenticating the search response from the BCM, the BLEM must determine which expected response to compare to the challenge response data ***FobCtlPssvData_No_Actl*** *(LBIFobCtrlSearchData)* but first, the BLEM shall verify that the received roll code matches the roll code that it sent to the BCM:

- If the roll code received does not match the roll code that the BLEM sent to the BCM, the BLEM shall ignore the response data ***FobCtlPssvData_No_Actl*** *(LBIFobCtrlSearchData)* and issue a new challenge request as defined by **[LBI.R384.001]**
- If the roll code received matches the roll code that the BLEM sent to the BCM, and if *LBIFobCtrlSearch_Rslt* = VALID (***FobCtlType_D_Stat = 0x2***), the BLEM shall verify the challenge response data received from the BCM matches bytes [0,4,9,10,13]
- If the roll code received matches the roll code that the BLEM sent to the BCM, and if *LBIFobCtrlSearch_Rslt* = INVALID (***FobCtlType_D_Stat = 0x1***), the BLEM shall verify the challenge response data received from the BCM matches bytes [1,2,6,8,11]

**[LBI.R390.001]** When the BLEM authenticates the search response and the search result is valid, then it shall report to the APIM that **a key has been found**.
When the BLEM authenticates the search response and the search result is invalid, then it shall report to the APIM that **a key has not been found**.

**[LBI.R391.001]** When the BLEM does not authenticate the search response, it shall attempt another key fob request to the BCM. (Only one re-attempt is allowed).

If the second attempt is not successful, then the BLEM shall not respond to the APIM's request for a key search. As a result, this will trigger an error message in the HMI.

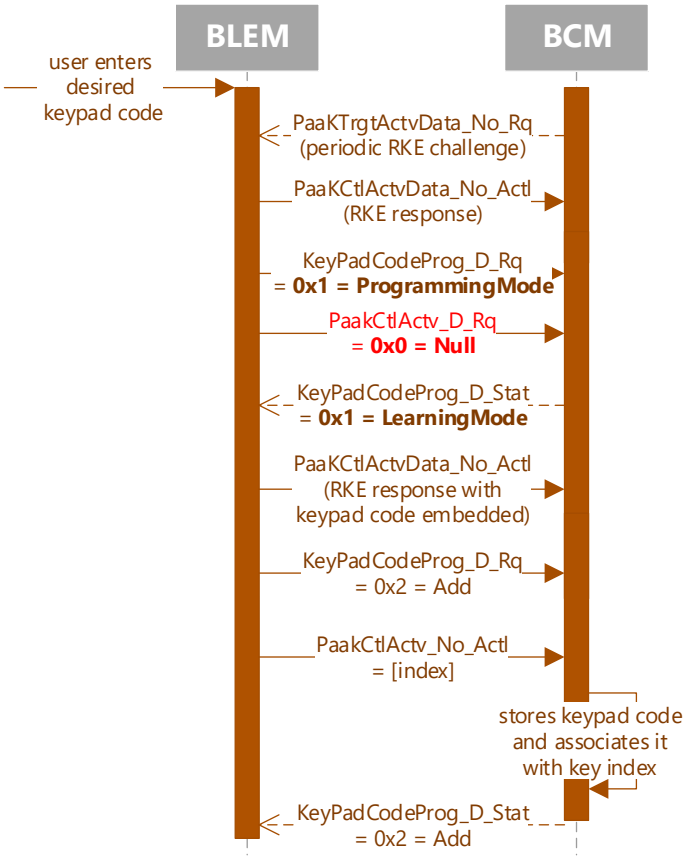
**[LBI.R392.001]** The BLEM shall wait for up to 2 seconds for a response from the BCM to a key search request. If the BCM does not respond with 2 seconds, then the BLEM shall not respond to the APIM's request for a key search. As a result, this will trigger an error message in the HMI.

**[LBI.R393.001]** When the BCM receives a search request with a search type *LBIFobTargetSearchType ≠ Registry*, it shall respond to the BLEM with Invalid search result *LBIFobCtrlSearch_Rslt*, challenge response = 0 (Null) and roll code = echo.

**[LBI.R394.001]** When the BCM receives a search request with search zone *LBIFobTargetSearchZone ≠ Interior*, it shall respond to the BLEM with Invalid search result *LBIFobCtrlSearch_Rslt*, challenge response = 0 (Null) and roll code = echo.

### 3.1.5 BLEM-BCM Interface requirements for Keypad programming

*Note: For this interface we are reusing the PaaK periodic challenge/response, see the PaaK SPSS **[PaaK-REQ-270046/A-Remote Start and RKE Challenge Response]**, where BLEM-BCM interaction is shown in below diagram:*



***PaakTrgtActvData_No_Rq*** *(PaakTargetRKEData)* is an array of 5 elements (40 bits) of challenge data sent by the BCM and used by the BLEM for calculating an authentication response.

***PaakCtlActvData_No_Actl*** *(PaakCtrlRKEData)* is an array of 5 elements (40 bits) of calculated authentication response by the BLEM.

The AES key shall be derived from the BLEM/BCM **TargetID** (see **PaaK SPSS requirement PaaK-REQ-242454**) and the AES input shall be derived from the challenge data ***PaakTrgtActvData_No_Rq*** *(PaakTargetRKEData)* as shown in **Figure 1 in Section 3.1.4.** The BLEM shall use this output to authenticate the challenge response from the BCM.

**[LBI.R395.001]** The BLEM shall respond to the ***PaakTrgtActvData_No_Rq*** (*PaakTargetRKEData)* BCM periodic RKE challenges using the signal ***PaakCtlActvData_No_Actl (****PaakCtrlRKEData).*

**[LBI.R396.001]** When the BLEM sends a request to enter keypad programming mode using an RKE response ***PaakCtlActvData_No_Actl*** *(PaakCtrlRKEData)* along with these signals:

> ***KeyPadCodeProg_D_Rq = 0x1*** (*ProgrammingMode),*
> ***PaakCtlActv_D_Rq*** (*PaakCtrlActionCode* = ***Null)***
> ***PaakCtlIndx1_No_Actl (index)***

***Note:*** *The index that the BLEM sends is the key index of the PaaK device that the entered password is associated with. The BLEM should set* ***PaakCtlIndx1_No_Actl = 1*** *and* ***PaakCtlIndx2-8_No_Actl = 0.***

The BLEM shall add ***PaakCtlActvData_No_Actl*** (*PaakCtrlRKEData)* request to the PaaK RKE Queue as described in **PaaK-REQ-270047:**

*"The PaakServer shall be able to queue up to three RKE commands.*
*The PaakServer shall transmit a queued command when new challenge data in PaakTargetRKEData is received in the order the commands were received.*
*If the queue is filled and a new RKE command is received, the PaakServer shall replace the oldest command with the newest command."*

When the BCM responds to a keypad programming request it shall set the signal ***KeyPadCodeProg_D_Stat*** = ***0x1*** (***LearningMode****)* and shall wait for another keypad action request from the BLEM to add LBI keypad code.

**[LBI.R397.001]** If the BLEM response to the BCM challenge is not valid, then the BCM shall timeout, set a DTC and report (***ProgrammingFailure****)* to the BLEM over CAN via the signal ***KeyPadCodeProg_D_Stat*** = ***0x5***.
If a keypad action code other than ***KeyPadCodeProg_D_Rq*** = ***0x1*** (***ProgrammingMode****)* is received by the BCM before successfully completing the authorization to enter keypad programming mode, the BCM shall report ***KeyPadCodeProg_D_Stat*** = ***0x5*** (***ProgrammingFailure****)* to the BLEM.

**[LBI.R398.001]** When the BLEM receives the signal ***KeyPadCodeProg_D_Stat = 0x1=*** *(LBIKeypadProg_Status) =* ***LearningMode*** confirming the BCM has entered programming mode, it shall send a keypad action request to add keypad code to the BCM using the signals:

> ***KeyPadCodeProg_D_Rq = 0x2*** *(LBIKeypadActionCode) =* ***Add,***
> ***PaakCtlActv_No_Actl = index*** *(PaakCtrl_RKESubID)*

The BLEM shall also send an RKE response with keypad code embedded within the signal ***PaakCtlActvData_No_Actl*** *(PaakCtrlRKEData)* least significant 2 bytes.

**[LBI.R399.001]** The BCM shall check the keypad code validity obtained from the BLEM.

- If the code is valid, the BCM shall store the keypad code into a code array, associate it with PaaK index, and report back to the BLEM that keypad code has been successfully added via the signal ***KeyPadCodeProg_D_Stat = 0x2*** *(LBIKeypadProg_Status) =* ***Add***
- If the BCM determines that keypad code is not valid, it shall report programming failure via the signal ***KeyPadCodeProg_D_Stat = 0x5*** *(Failed) =* ***ProgrammingFailure*** and stay in programming mode. The BLEM shall then need to re-request the *Add* action with a different keypad code (e.g. the user must be informed to try again with a different keypad code).
- If the number of attempts from the BLEM exceeds a threshold of 3 (1 original plus 2 attempts), the BCM shall accept a duplicate keypad code and store it.

**[LBI.R400.001]** When requesting to delete a single keypad code, the BLEM shall provide the Paak Index of the keypad code that is to be deleted.
The BCM shall then erase the keypad code from the memory, and report its successful deletion via the signal ***KeyPadCodeProg_D_Stat = 0x3*** (*Delete) =* ***Delete*** to the BLEM.

**[LBI.R401.001]** For both the *Add* and *Delete* operations *(**KeyPadCodeProg_D_Rq = 0x2** & **0x3** (LBIKeypadActionCode) = **Add** & **Delete**)*, the BLEM shall reserve a Paak Index of 63 for the Valet keypad Code.

**[LBI.R402.001]** When requesting to delete all LBI keypad codes, the BCM shall erase all keypad codes stored in memory, and report its successful deletion via the signal ***KeyPadCodeProg_D_Stat = 0x4*** (*Delete_All) =* ***DeleteAll*** to the BLEM.

**[LBI.R403.001]** After the BCM has performed the requested operation or has reported that operation is failed, the BCM shall wait 1000 msec and then exit programming mode and return to normal mode ***KeyPadCodeProg_D_Stat = 0x0 = NormalMode***, unless the keypad code was determined as invalid. In that case, the BCM shall return to learning mode ***KeyPadCodeProg_D_Stat = 0x1 = LearningMode*** and wait for the next keypad action code from the BLEM.
If no keypad action code *(Add/Delete/DeleteAll)* is received from the BLEM after entering learning mode ***KeyPadCodeProg_D_Stat = 0x1 = LearningMode*** (including after an invalid keypad code was detected on an *Add* operation), the BCM shall return to normal mode ***KeyPadCodeProg_D_Stat = 0x0 = NormalMode*** within 2000 msec.

## 3.2 HMI Requirements

**[LBI.R297.01]** Any time SYNC displays a list of PaaK devices, the list shall be sorted by key index, with the lowest index at the top.

### 3.2.1 PaaK Backup Settings

**[LBI.R130.01]** SYNC shall provide a menu for PaaK Backup settings when the vehicle is configured for Phone-as-a-Key:

| | | | | | | |
|---|---|---|---|---|---|---|
| DE05 | 1 | 3 | Phone as a Key | 0 | 0 – Not Present<br>1 – Present | N/A |

**[LBI.R131.03]** The PaaK Backup settings menu shall provide buttons for the following functions:

- Create backup passwords/keypad codes for phones-as-keys
- Reset backup passwords/keypad codes for phones-as-keys
- Delete backup passwords/keypad codes for phones-as-keys

**[LBI.R132.03]** SYNC shall not allow the user to initiate password creation, deletion, or reset from the HMI unless the ignition is in Run (*Ignition_Status = 0x4 = Run*) and transmission is in Park (*GearLvlPos_D_Actl = 0x0 = Park*). The buttons to initiate these functions shall be greyed out unless these conditions are met.

**[LBI.R373.01]** If password creation, deletion, or reset are active in SYNC (e.g. user is in the process of creating a backup password), and the ignition changes from Run or the transmission changes from Park, then SYNC shall display a message with driving restriction information, exit the function and return to the PaaK Backup settings menu.

### 3.2.2 Enhanced Valet Mode

**[LBI.R374.01]** SYNC shall not allow the user to activate Enhanced Valet Mode unless the ignition is in Run (*Ignition_Status = 0x4 = Run*) and transmission is in Park (*GearLvlPos_D_Actl = 0x0 = Park*). Ignition and transmission state shall be checked when the user selects the Valet Mode button from the SYNC HMI. If ignition is not in Run and transmission is not in Park, then a message with driving restriction information shall be displayed.

*Note: This requirement assumes that PaaK is enabled and a backup password has been created.*

**[LBI.R375.01]** If the password entry screen for activating Enhanced Valet mode is active in SYNC and the ignition changes from Run or the transmission changes from Park, then SYNC shall display a message with driving restriction information, exit the function and return to SYNC Settings menu.

**[LBI.R133.02]** When Enhanced Valet Mode is activated, the SYNC HMI shall display the generated valet password on screen until SYNC suspends (i.e. ignition is switched off and user opens door). After SYNC suspends, this valet password shall not be displayed in the SYNC HMI again.

**[LBI.R134.04]** When Enhanced Valet Mode is active, the SYNC HMI shall provide a valet-specific screen for entering the valet password when a valid key is not detected at startup. This screen shall be distinguishable from the screen for entering the backup password.

**[LBI.R135.03]** When SYNC is on and Enhanced Valet Mode is active, the SYNC HMI shall indicate that the vehicle is in Valet Mode and provide an option to exit/deactivate. No other parts of the HMI shall be accessible.

**[LBI.R376.01]** SYNC shall not allow the user to exit/deactivate Enhanced Valet Mode unless the ignition is in Run (*Ignition_Status = 0x4 = Run*) and transmission is in Park (*GearLvlPos_D_Actl = 0x0 = Park*). The button to initiate this function shall be greyed out unless these conditions are met.

**[LBI.R136.02]** When the option to exit Enhanced Valet Mode is selected and the vehicle detects a key (PaaK device or key fob) in the vehicle, the SYNC HMI shall exit Enhanced Valet Mode and return to the previous state. If a key is not detected in the vehicle, the backup password entry screen shall be displayed. If the user enters a valid backup password at this screen, the SYNC HMI shall exit Enhanced Valet Mode and return to previous state.

**[LBI.R377.01]** If the password entry screen for deactivating Enhanced Valet mode is active in SYNC and the ignition status changes from Run or the transmission changes from Park, then SYNC shall display a message with driving restriction information, exit this function and return to the screen indicating that the vehicle is in Valet Mode.

### 3.2.3 Notifications

**[LBI.R137.03]** Whenever a user creates, starts the vehicle with, deletes, or resets a backup password, a notification shall be sent to the user that the backup password is associated with. This notification shall be sent as a push notification to the user's PaaK device and as an email to the user's email account (where available).

**[LBI.R179.03]** Whenever a user authorizes creation of a valet password via the presence of PaaK device(s) in the vehicle, the valet password shall be delivered via Bluetooth as a push notification to all PaaK devices detected in the vehicle. No other notification shall be delivered to the user.

**[LBI.R180.02]** Whenever a user authorizes creation of a valet password via a backup password, a notification shall be sent to the user that the backup password is associated with. This notification shall be sent via push notification to user's PaaK device and as an email to the user's email account (where available).

**[LBI.R378.01]** Whenever a user starts the vehicle with a valet password, a notification shall be sent to all users that received the valet password (if valet password creation was authorized by device(s)) or to the user that the backup password is associated with (if valet password creation was authorized by a backup password). This notification shall be sent via push notification to user's PaaK device and as an email to the user's email account (where available).

**[LBI.R181.01]** Whenever a user deletes a valet password (i.e. exits Enhanced Valet Mode), a notification shall be sent to all users that received the valet password (if valet password creation was authorized by device(s)) or to the user that the backup password is associated with (if valet password creation was authorized by a backup password). This notification shall be sent via push notification to user's PaaK device and as an email to the user's email account (where available).

**[LBI.R182.01]** Whenever a user is locked out from PaaK Backup, a notification shall be sent to all users of PaaK Backup. This notification shall be sent via push notification to user's PaaK device and as an email to the user's email account (where available).

**[LBI.R381.02]** Whenever a user disables Vehicle Connectivity via the Customer Connectivity Settings (CCS) in SYNC HMI, the PaaK BSP (Backup Starting Passcode aka LBI) feature shall be disabled.

**[LBI.R404.01]** Prior to disabling Vehicle Connectivity via the Customer Connectivity Settings (CCS) in SYNC, the SYNC HMI must inform a user that if Connectivity is turned off, the PaaK BSP feature shall be disabled and a user will not be able to start the vehicle without a valid key. If a user acknowledges, then SYNC HMI must grey out all associated PaaK BSP (Backup Starting Passcode aka LBI) feature settings.

***Note:*** *When the Vehicle Connectivity is OFF, a customer should still be allowed to use default factory keypad code, any previously created personal keypad and the PaaK BSP keypad codes, as well as classic valet. When Vehicle Connectivity becomes enabled, the previously greyed out PaaK BSP feature settings in SYNC HMI will be selectable again and all suspended passcodes will be active.*

## 3.3 Security Requirements

**[LBI.R138.01]** Passwords shall never be transmitted across any interface in clear-text.

### 3.3.1 Password Types

**[LBI.R139.01]** The BLEM shall recognize two different types of passwords: a “backup” password and a “temporary” password.

*Backup passwords are intended for long-term use, while temporary passwords are short-term passwords intended for usage in valet-type scenarios.*

### 3.3.2 Password Storage

**[LBI.R140.01]** The BLEM shall securely store within HSM all customer passwords created and used to enable vehicle start and drive-away.

**[LBI.R141.01]** Passwords shall never be stored in the clear, but instead shall be stored in a salted and hashed format, defined as:

$$Programmed\ Hash = SHA256(Salt + Password)$$

*Note that the output of SHA256 will always be a 256-bit (32-byte) hash – thus storage requirements are consistent regardless of password length.*

**[LBI.R0142.01]** Upon completing provisioning, the BLEM shall randomly generate a 128-bit salt. This salt may be stored in unsecured memory (i.e. outside HSM).

### 3.3.3 Backup Passwords

#### *3.3.3.1 Programming*

**[LBI.R143.01]** The BLEM shall not accept any new backup passwords for LBI unless all of the following conditions have been met:

- The customer has opted-in to using the feature
- At least one PaaK device authorized and enabled for the given vehicle (i.e. with a provisioned CAK) is inside the vehicle and is in session (see BLE Interface Security Specification)
- A password does not already exist for the given PaaK device/vehicle pairing
- At least one PEPS key fob is detected inside the vehicle

**[LBI.R144.02]** PaaK device checks and key fob checks shall be executed when the BLEM has received a password hash to be stored.

*This mitigates potential time of check/time of use vulnerabilities.*

**[LBI.R145.01]** All backup passwords MUST be associated internally on the BLEM to a CAK authorized for the given vehicle.

**[LBI.R146.03] For CGEA1.3C** architecture with SYNC 3/ or Feature Bundle 4 (FB4) vehicle programs the **passwords shall be a minimum of 5 characters** in length if a mixture of letters, numbers, and symbols are used.
**For FNV2** architecture with SYNC 4/ or Feature Bundle 5 (FB5) vehicle programs the **passwords shall be a minimum of 6 characters** in length if a mixture of letters, numbers, and symbols are used.
**For CGEA1.3C** architecture with SYNC 3/ or Feature Bundle 4 (FB4) vehicle programs the **passwords shall be a minimum of 8 characters** in length if only numbers are used (for example, Enhanced Valet passcode).
**For FNV2** architecture with SYNC 4/ or Feature Bundle 5 (FB5) vehicle programs for European market, the **passwords shall be a minimum of 10 characters** in length if only numbers are used (for example, Enhanced Valet Passcode).

**[LBI.R147.01]** Passwords shall be a maximum of 64 characters in length.

**[LBI.R148.01]** Backup passwords shall not be directly displayed on any HMI.

**[LBI.R149.01]** An indicator shall be displayed on the HMI indicating the relative strength of the user's selected password.

**[LBI.R150.02]** The strength indicator shall have four levels of strengths: weak, fair, good, and strong. These levels shall be indicated using a four-segment fill bar. If minimum requirements are not met, bar shall be empty. If password is weak, bar shall fill ¼ with red color. If password is fair, bar shall fill ½ with orange color. If password is good, bar shall fill ¾ with yellow color. If password is strong, bar shall fill completely with green color.

**[LBI.R178.02]** The following rules shall be used to determine back up password strength:

1. Weak: Password must have at least eight (8) characters if password consists only of numbers or at least five (5) characters if password does not consist only of numbers.
2. Fair: Password must have at least eight (8) characters including at least one (1) lower-case letter, one (1) upper-case letter, and one (1) number.
3. Good: Password must have at least ten (10) characters including three of the following four types of characters: lower-case letter, upper-case letter, number, special character (including space). Password also must have not more than two identical characters in a row.
4. Strong: Password must have at least twelve (12) characters including three of the following four types of characters: lower-case letter, upper-case letter, number, special character (including space). Password also must have not more than two identical characters in a row.

- The SYNC shall utilize password strength controls according to the following security guidelines:

https://www.owasp.org/index.php/Authentication_Cheat_Sheet#Implement_Proper_Password_Strength_Controls

- The SYNC must use at the minimum 10K common password list and if the password is found in that list the ranking shall be marked not higher than "fair":

https://github.com/danielmiessler/SecLists/tree/master/Passwords

- A password that does contain two (2) or more similar letters/numbers shall be excluded from the "strong" or "good" ranking.
- A password that does not contain two (2) or more similar letters/numbers shall be eligible for the "strong" or "good" ranking.

*For the following requirements describing hash calculation, the '+' operand is construed to mean concatenation, not addition.*

**[LBI.R151.02]** Backup passwords shall be transmitted from the in-vehicle HMI in a salted and

hashed format (“Programmed Hash”), using the following mechanism:

$$Programmed\ Hash = SHA256(Salt + Password)$$

***Note:*** *the “+” operator in above calculations means append, not add. The salt and nonce values must be a lower case hex bytes. There is no space between (salt and password) string, it is one string of hex values. SYNC must convert the password that user is trying to create from asci value into hex string before proceeding with Programmed Hash calculation.*

*Calculation Example*

**Programmed hash** = SHA256(e7e272cf7bbf00ab9874ad03bde24626717765313233) = 35effe7d0c505913286470ba328f835da9067cd1c9ca2817102db7ec6f95b4c8

*Where*

**Salt** = e7e272cf7bbf00ab9874ad03bde24626

**Desired password** = qwe123 → converted to hex = 717765313233

**[LBI.R152.01]** If more than one PaaK device is discovered within the vehicle without an associated backup password, the user shall be prompted to select a device to associate the password with.

#### *3.3.3.2 Deletion*

**[LBI.R153.01]** Customers shall be required to enter the backup password to delete it via the in vehicle HMI.

**[LBI.R154.01]** If a CAK is revoked for any reason and a backup password is associated to that CAK, the backup password shall immediately be deleted and not allowed for further usage.

**[LBI.R155.01]** Backup passwords shall only be deleted if the vehicle is in a motive mode (i.e. an authentication step has been performed enabling the user to start the vehicle, either by PaaK device, key fob or backup password).

### 3.3.4 Password Usage

*Note: the procedure described below applies for both temporary and backup passwords.*

**[LBI.R156.01]** A challenge/response mechanism shall be used with the password hash to authenticate and start the vehicle via the in-vehicle HMI.

**[LBI.R157.01]** On request, the BLEM shall generate a random 256-bit nonce using the best available RNG. If possible, a true RNG shall be used. This nonce shall be sent to the in-vehicle HMI module.

**[LBI.R158.02]** When the user enters a password, the in-vehicle HMI module shall perform two rounds of hashing before transmitting the response to the BLEM. The first shall calculate the Programmed Hash:

$$Programmed\ Hash = SHA256(Salt + Backup\ Password)$$

***Note:*** *see notes for LBI.R151.02*

**[LBI.R159.02]** Once the Programmed Hash has been calculated, the in-vehicle HMI module shall then calculate and transmit the "Authentication Hash", using the following mechanism:

$$Authentication\ Hash = SHA256(Nonce + Programmed\ Hash)$$

***Note:*** *the "+" operator in above calculations means append, not add. The nonce value must be a lower case hex byte. There is no space between (nonce and programmed hash) string, it is one string of hex values.*

*Calculation Example*

**Authentication hash** = SHA256(fb03845b994809d8b265aba4a7c7c64235d125151d7f1eb3fcf42252148a483435effe7d0c505913286470ba328f835da9067cd1c9ca2817102db7ec6f95b4c8) = **7a86e7379848b4106150e3674aaf0f99a1b408588d8b402742782e6db0d16c3e**

Where

**Nonce** = fb03845b994809d8b265aba4a7c7c64235d125151d7f1eb3fcf42252148a4834

**Programmed Hash** = 35effe7d0c505913286470ba328f835da9067cd1c9ca2817102db7ec6f95b4c8

Refer to APIM / BLEM TP specs (BackupIgnition_Rq and BackupIgnition_Rsp) for more details on where to apply Programmed hash and where to apply Authentication hash.

BLEM and Sync both shall validate the output of Programmed hash or Authentication hash depending upon which opcode they are comparing.

**[LBI.R160.01]** Upon sending the nonce to the in-vehicle HMI module, the BLEM shall also calculate the Authentication Hash for all stored passwords.

*This is done because the BLEM does not know ahead of time which password to expect, and thus it must compare the received value to all known passwords to find a match.*

**[LBI.R161.01]** Upon receiving the Authentication Hash, the BLEM shall compare the received values to its calculated Authentication Hashes. A constant time algorithm shall be used when comparing the hashes. If a match is found, the system shall indicate success to the user and allow vehicle start. If a match is not found, a failure should be reported to the user.

**[LBI.R162.01]** If the user does not provide a valid password within 5 attempts, the system shall lock out further attempts for 5 minutes. The authenticating system (the BLEM) shall be the master of this logic.

**[LBI.R163.01]** After the first 5 minute lockout, the user shall be permitted 5 additional attempts to enter a valid password. If the user does not enter a valid password within these 5 attempts, the system shall lock out further attempts for another 5 minutes. This lockout process shall repeat indefinitely.

### 3.3.5 Temporary Passwords

Short-term passwords may also be enabled by the user for valet scenarios.

**[LBI.R165.02]** The user shall be required to have a PaaK device with an associated password in the vehicle or enter a valid backup password to enable a temporary password.

**[LBI.R174.02]** The Temporary Password Hash shall be valid until the system exits Enhanced Valet Mode.

**[LBI.R175.02]** The system shall require a valid backup password, PaaK-authenticated device in vehicle, or key fob in vehicle, together with confirmation from the user to exit Enhanced Valet Mode.

### 3.3.6 Notifications

**[LBI.R176.04]** Upon creation, successful usage, deletion, or reset of any backup or valet password, the BLEM shall send a SyncP signed message (Service Type 0x40/Sub-Service 0x0) to the SDN via TCU. The payload shall contain:

- The Key ID of the associated CAK (or CAK(s) for valet password creation)
- The timestamp of the action
- An indicator of whether a password was created, successfully used, deleted, or reset
- An indicator of whether the event was for a backup password or a valet password
- An indicator of whether a password or PaaK device authorized valet password creation
- An indicator of whether a password, key fob, PaaK device, or both a key fob and PaaK device authorized valet password deletion

**[LBI.R382.01]** When a user successfully follows through with a Master or PaaK Reset, the BLEM shall not send SyncP signed message (Service Type 0x40/Sub-Service 0x0) to the SDN via TCU.

**[LBI.R177.03]** Upon lockout of the system, the BLEM shall send a SyncP signed message (Service Type 0x40/Sub-Service 0x1) to the SDN via TCU. The payload shall contain an indicator of whether or not valet mode was active at the time

*Note: The TCU does not process these SyncP messages. It timestamps and relays them to the SDN.*

**[LBI.R329.01]** When the SDN receives a SyncP message from the BLEM via TCU, it shall verify that the SyncP signature is correct (by BLEM ESN) and then forward this message to PaaK FI for processing.

**[LBI.R300.02]** PaaK FI shall keep a history of these events for one year.

# 4 Appendix A: Definitions / Acronyms

| Acronym | Full Name | Description |
|---|---|---|
| APIM | Accessory Protocol Interface Module | Also known as “SYNC module” |
| BCM | Body Control Module | |
| BLEM | Bluetooth Low Energy Module | Main controller for Phone-as-a-Key system |
| CAK | Consumer Access Key | One of the virtual keys delivered to the phone or the vehicle. Used for authorizing phones to the vehicle. |
| CAN | Controller Area Network | |
| IPC | Instrument Panel Cluster | Also known as “cluster” |
| LBI | Lincoln Backup Ignition | a.k.a. PaaK Backup. Backup starting feature tied to Phone-as-a-Key. |
| PaaK | Phone-as-a-Key | Feature whereby user is able to enter and start vehicle with mobile phone similarly to PEPS. |
| PaaK Backup | Phone-as-a-Key Backup | a.k.a. Lincoln Backup Ignition<br>Backup starting feature tied to Phone-as-a-Key. |
| PaaK BSP | Phone-as-a-Key Backup Starting passcode | a.k.a. Lincoln Backup Ignition<br>Backup starting feature tied to Phone-as-a-Key. |
| PEPS | Passive Entry, Passive Start | In the context of key fobs, a feature that allows user to enter and start the vehicle without interacting with their key fob i.e. passive operation. |
| TCU | Telematics Control Unit | |
| SDN | Service Delivery Network | Also known as the “cloud” or the “back-end”. |

# 5 Appendix B: Reference Documents

| Reference | Title | Doc. ID | Revision |
|---|---|---|---|
| 1 | Phone-as-a-Key PSD | | 1.3 |
| 2 | BLE Interface Security Specification | | 1.0.1 |
| 3 | TP BLEM SPSS | | 1.2 |
| 4 | BLEM/BLEAM Common Function BLEM SPSS | | 1.1 |
| 5 | Phone-as-a-Key BLE Communication Protocol | | V5 |
| 6 | H21 SYNC 3 GUI Design Standards | | 3.02 |
| | | | |
| | | | |
| | | | |
| | | | |