Backup and Restore Feature in RPI - Design

1. Introduction

Backup and Restore is the feature developed in regard to WebUI.

The user can Backup the current user settings to the local PC and Restore the same to device when required.

Backup Settings:

- Allows the user to save the current settings
- Settings from database files are backed up
- The backup files are encrypted using the secure key inputted from the User
- Generates a tar file of the encrypted database files
- Saves the tar file to Local PC.

Restore Settings:

- Allows the user to select the backup file with required settings
- Extracts the tar file containing database files
- Decrypts the database files using valid secure key entered by user
- Loads backed up settings to device
- Device will be rebooted automatically to apply the settings.

2. Design Considerations

Backup and Restore functionalities are supported by a set of script files. Following are the mentioned script files:

ConfPhp

download_user_settings.php

upload_user_settings1.php

upload_user_settings2.php

backup_user_settings.php

backup_enc_key.php

ajax_at_saving_backup_key.php

These script files which are already present in the RDK build, will execute when the Backup and Restore buttons are clicked.

2.1. Backup Settings

Scripts **backup_user_settings.php, download_user_settings.php** downloads the tar file containing database files**.**

The database files which contains the Current User settings are used for back up. These database files are encrypted using the secure key entered by the User, scripts executed are **backup_enc_key.php** and **ajax_at_saving_backup_key.php**.

A tar file is generated with these encrypted files together and stored in temporary folder **/tmp/**. The tar file from temporary folder is downloaded to the Local PC of User.

2.2. Restore Settings

Scripts **upload_user_settings1.php, upload_user_settings2.php** perform the Restore and Status update functions.

The files which are uploaded in WebUI are extracted to get the encrypted database files. These database files are decrypted using the valid secure key entered by user.The secure key entered by user should be the same secure key which was entered earlier during encryption. If it is invalid, the decryption does not happen and restore fails.

The database files are saved to temporary folder **/var/tmp/**. The files are retrieved and replaced with the existing database files in RPi image.

Other Considerations (DAR)

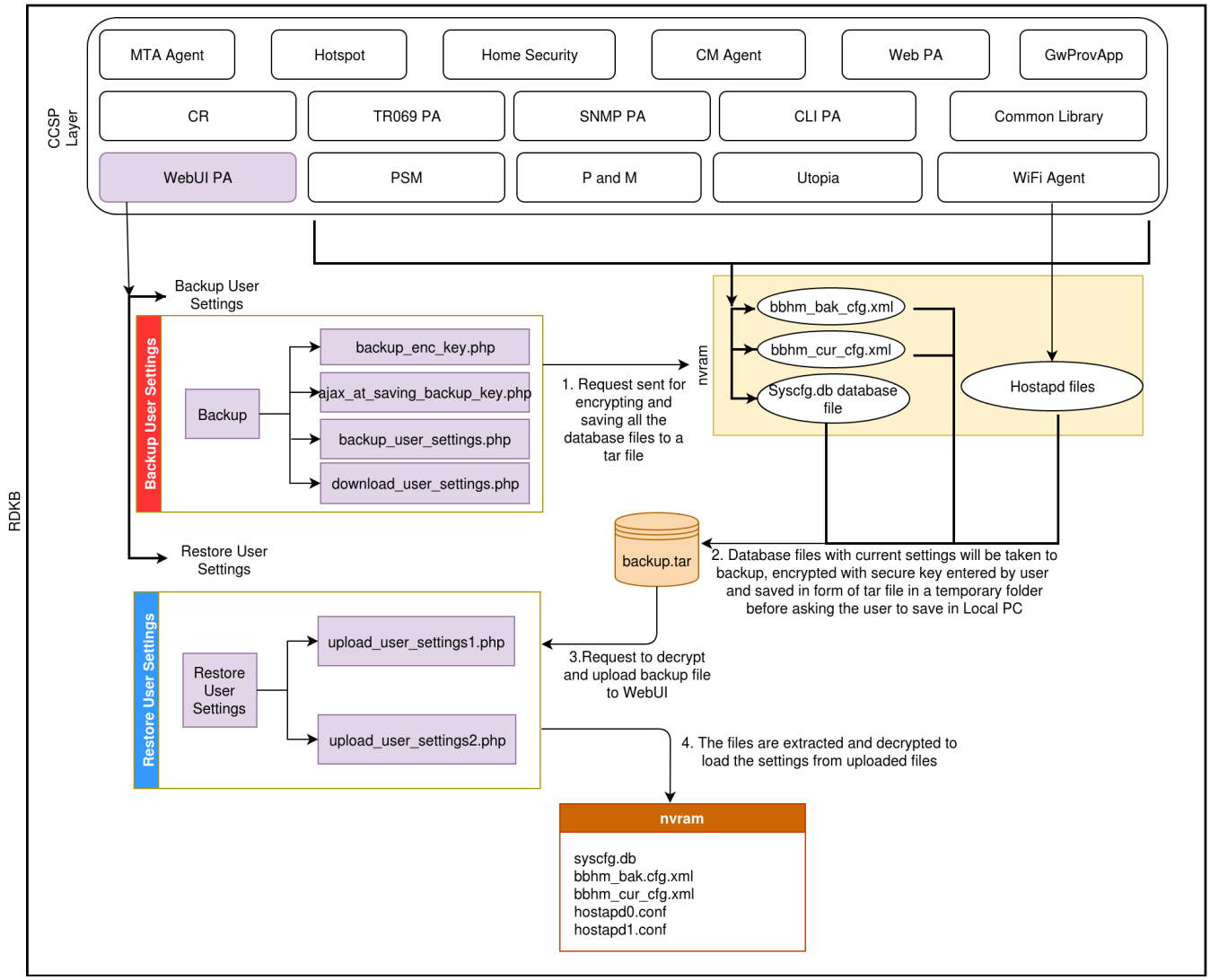
It is required for the database files to be saved in an encrypted format to maintain the security of data, after backup.

During code development, two approaches were considered while designing the encryption/decryption

- **1st Approach:** During Backup, the database files need to be encrypted to avoid corruption of data. The password used for user login to WebUI was considered as the secure key for encrypting the database files. This password is saved in the database files by default (though in the form of hashed password). The password has to be encoded and saved in a file to use for decryption during restore, to avoid conflict when restore is done after changing password for logging in WebUI. This causes high risk of security.
- **2nd Approach:** During Backup, the user is asked to enter the secure key, which is used to encrypt the files. After encryption, during restore the user has to enter the same secure key to decrypt. If the secure key is invalid, a error message will be returned. at this point, the decryption does not happen and restore fails. This secure key is not saved anywhere.

The latter approach has been implemented as it was found to be more preferable and acceptable as compared to the former. This is because the former approach causes risk of security as the password is accessible by user using database file or encoded file.

3. **Architecture**



4. **Limitations**

The downloaded file, **backup.tar** is the default name given to the backup file. Once downloaded to Local PC, this file has to be replaced with new backup. tar file. If the old file is not deleted, it may cause misbehavior in Restore feature execution. The old file can be either deleted or renamed to avoid duplicate file creation. Please Refer **User manual** for further details.

During Restore, the file to be uploaded has to be named to **backup.tar** by default.

5. Future Enhancements

5.1. Backup Feature

- Naming convention for the backup.tar file with versions/time of file creation can be automated.

  Example: backup_2711191230.tar for files generated on 27/11/19 12:30PM.

5.2. Restore Feature

- During file upload, selection of files with any name, can be made possible.

  Example: Wifi_settings.tar