USER GUIDE

Cyber Reliant Defender Installation and Use

Mar 2023

CYBER RELIANT

Table of Contents

Table of Figures

1. General Information

1.1. Introduction

This document will describe the installation and use of the Cyber Reliant Defender management service. This Application installation and Authentication are necessary for the use of the **CRC Data Defender Library.**

Version:

Cyber Reliant Defender management service	**Version 4.0**
Cyber Reliant Data Defender Library	**Version 4.0**

1.2. Target Devices

See Section 1.3 of the Security Target for a list of mobile devices included in the scope of evaluation for the Cyber Reliant Defender.

1.3. Management Service & Device Administration Tool

Cyber Reliant Defender Management Service is a separately installed Android application ("CRC Defender"). Its purpose is to provide a second layer of user authentication (dual Auth) and to create, manage and distribute the encryption keys (FEKEK) that are required by the **CRC Data Defender Library**. The Management Service is pre-configured before installation with settings necessary for proper use and registration of the CRC Plugin.

Below is the icon:

2. Critical Bug Reporting & Patch Versions

2.1. Version Updates

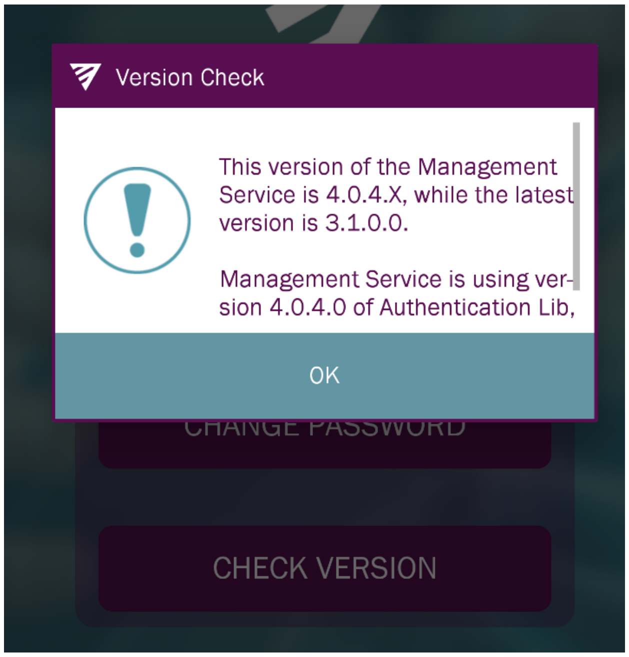
CRC will email all customers using outdated software as soon as the new version release is available. CRC will also list the current software version numbers on their website (www.cyberreliant.com).

Customers are able to request the newest updated versions by emailing their request to: contact@cyberreliant.com .

Upon a customer request CRC sends new software release version and Release Notes to the customer, usually via email, containing a download link to the customer installable software version .apk file hosted by CRC's ShareFile portal.

2.1.1. Version checking on device

Users can check their current installation version and the last Full release version using the Management Service Main window (see section 4.2).



Version Check

This version of the Management Service is 4.0.4.X, while the latest version is 3.1.0.0.

Management Service is using version 4.0.4.0 of Authentication Lib,

Note that the 'latest version' listed in the UI is the most recent commercial released version. The version installed may be a more recent version that has not yet been commercially released.

2.2. Reporting a bug

If you believe you have discovered a bug with the software, please follow the steps below:

1. Document the steps required to reproduce the bug
2. Send an encrypted email to support@cyberreliant.com containing the following information:
    a. Name
    b. Contact information
    c. Version of software being used
    d. Description of the bug
    e. Steps to reproduce
    f. Any screenshots, if applicable

For further assistance how to send an encrypted email, see the following link:

https://support.microsoft.com/en-us/office/encrypt-email-messages-373339cb-bf1a-4509-b296-802a39d801dc

2.3. CRC response to reported bugs

CRC will respond to your email acknowledging that the submitted bug has been received and that it is being reviewed. CRC will then attempt to reproduce the bug internally and assign a severity level to the bug. The severity levels are as follows:

Critical (S1): crashes product, feature unusable, a severe security risk, etc.

Major (S2): significant defect, issue with behavior, does not function as intended

Minor (S3): undesirable behavior, may have acceptable workaround

Low (S4): minor issue (text, trivial UI issue, etc.)

CRC will then inform the bug reporter of the severity level, any workarounds for the bug, and if a software patch will be created to fix the bug.

2.4. Software patch

If a critical or security-relevant bug is discovered, CRC will release a software update no later than 30 business days after acknowledging the critical bug. The process below lists the steps to releasing a software update

1. CRC Engineering reviews bug and estimates time to fix
2. CRC Engineering fixes bug
3. CRC Engineering tests their fix to ensure it works
4. CRC Engineering builds new software patch candidate and distributes internally to conduct quality assurance (QA)
5. CRC QA tests the software patch candidate to ensure the fix works
6. CRC QA notifies Engineering that the fix works
7. CRC QA performs a final test of the software, testing various features to ensure nothing else broke in the patching process
8. CRC updates product Release Notes with new version number, documentation of the bug that was fixed, and if any changes have been made to security properties or the configuration
9. CRC sends new software release version and Release Notes to the customer, usually via email, containing a download link to the customer installable software version .apk file hosted by CRC's ShareFile portal.

Note: The above process may contain loops. For example, if step 5 fails, the process returns to step 2. It is also possible that fixing the bug can break other aspects of the software (failure of step 7). If this happens, the new failures will be assessed on their severity levels and fixed accordingly. Allocating 30 business days for fixing ensures that there is plenty of time to perfect the patch in case the bug is more complex than initially planned. CRC will work with the mentality to fix critical bugs and test as soon as possible, and only use the full 30 business days if absolutely necessary.

2.5. New Patch Announcements

CRC will email all customers using outdated software as soon as the new release is available. CRC will also list the current software version numbers on their website (www.cyberreliant.com).

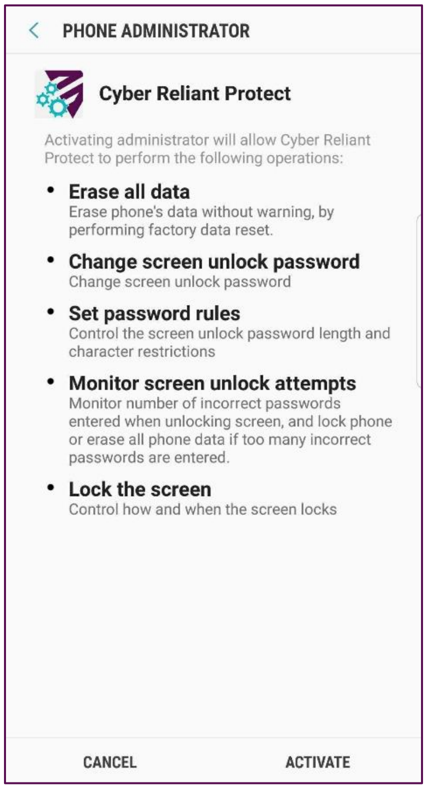
2.6. Notifications of new critical bugs

If a critical bug is found, all customers using the affected software version will be notified via email about the bug and will be informed about the upcoming patch release.

3. Installation

To install the Cyber Reliant APK, follow the steps below:

1. Download the .apk file to the device.
2. Open the .apk file.
3. Tap **Install**.
4. Once installation is complete, open the **CRC Protect** app. An "Activate Phone Administrator" screen will appear:


PHONE ADMINISTRATOR

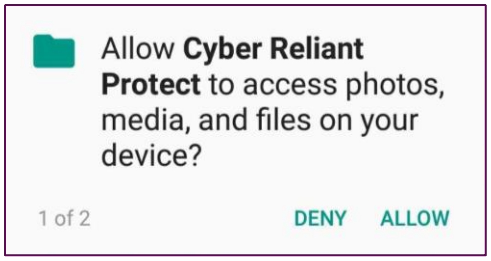
Cyber Reliant Protect

Activating administrator will allow Cyber Reliant Protect to perform the following operations:

- **Erase all data**
  Erase phone's data without warning, by performing factory data reset.
- **Change screen unlock password**
  Change screen unlock password
- **Set password rules**
  Control the screen unlock password length and character restrictions
- **Monitor screen unlock attempts**
  Monitor number of incorrect passwords entered when unlocking screen, and lock phone or erase all phone data if too many incorrect passwords are entered.
- **Lock the screen**
  Control how and when the screen locks

Figure 1: Activate Phone Administrator screen

5. Tap **ACTIVATE**. A confirmation popup will appear:



Allow **Cyber Reliant Protect** to access photos, media, and files on your device?

1 of 2 DENY ALLOW

Figure 2: Confirmation of Phone Administrator popup

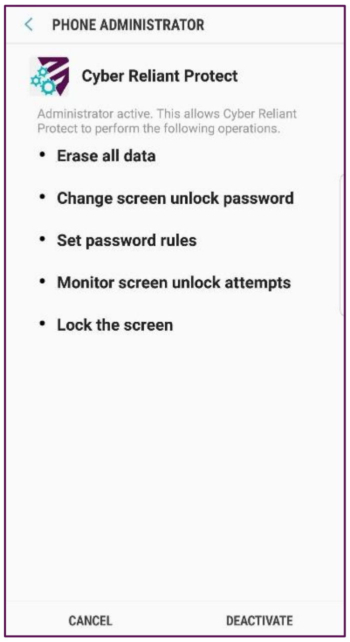
6. Tap **ALLOW**.
7.

3.1. Uninstallation

WARNING! If Uninstalled, all encrypted files will be inaccessible going forward with no possibility of recovery!

All encrypted files will become inaccessible once uninstallation is complete. To uninstall Cyber Reliant Protect, use the following steps:

1. Navigate to Android's settings screen.
2. Navigate to security settings.
3. Locate the **Device administrators** setting.
4. On the **Device administrators** screen, locate the Apps page and select the **Cyber Reliant Defender** App.
5. Tap **UNINSTALL**. A confirmation popup will appear.



PHONE ADMINISTRATOR

Cyber Reliant Protect

Administrator active. This allows Cyber Reliant Protect to perform the following operations.

- Erase all data
- Change screen unlock password
- Set password rules
- Monitor screen unlock attempts
- Lock the screen

CANCEL DEACTIVATE

6. Tap **Deactivate**.

4. Overview & Elements

4.1. Change Password Screen

Once the Management service is started, it will Load all of the embedded configuration values into the system and present the user with the initial 'Change Password' screen:

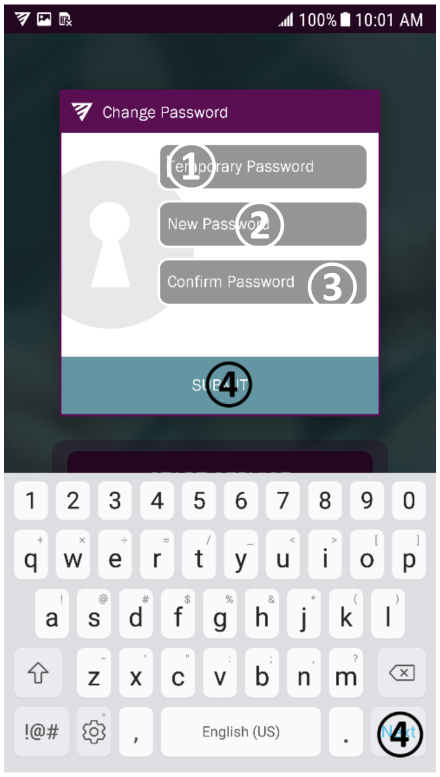


Figure 3: Home Screen (no configuration)

Field	Description	Input Type
1. Default Password	Initial Password set in embedded config. Use: **DEFENDER**	text
2. New Password	User selected password (min 6 chars)	text
3. Confirm password	Repeat input of User selected password for verification.	text
4. Submit	Submits password change	button

The Management Service installation has an embedded configuration the sets a Default temporary password on install. That password must be changed to a User selected password on startup. That is the password that will be used going forward for all future authentication of the CRC system.

In this case, the system will only require authentication at each Boot of the device.

Though the password criteria for minimum number of characters, upper, lower and special characters is specified in the configuration settings of the installation it is up to the user to select a strong passphrase.

For a user to create a strong passphrase, a passphrase with at least one uppercase letter, lowercase letter, special character, and number is recommended. Longer length passphrases provide increased security strength over shorter ones. Acceptable password strength should reflect the sensitivity of the data being encrypted. Higher data sensitivity should have increased password strength.

4.2. Start Service:

Once the newly created User password is created and submitted, The first of the Management service screens will be displayed.

The Start Service page:

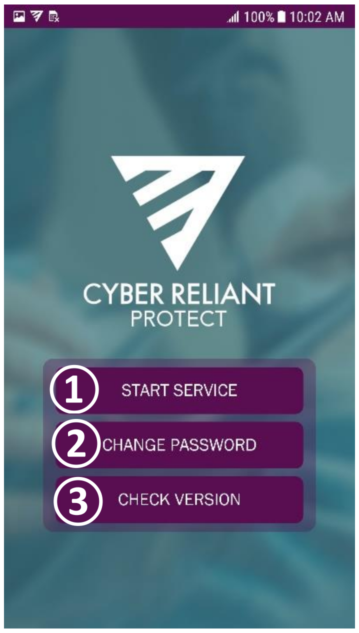


Figure 4 Start Service

Field	Description	Input Type
1. Start Service	Starts the Management Service with the user password and embedded configuration	Button
2. Change Password	Allows the user to change password as above.	Button
3. Check Version	Performs a version check of Cyber Reliant Protect to determine if the installed version is up-to-date. Internet connection required.	Button

Selection of the ‘Start Service’ button will send the full configuration to the Management Service for processing.

4.3. Authentication:

Once the Start Service has been selected, all user and start up configuration is finished and so the Management service will need it's first authentication using the newly created User password.

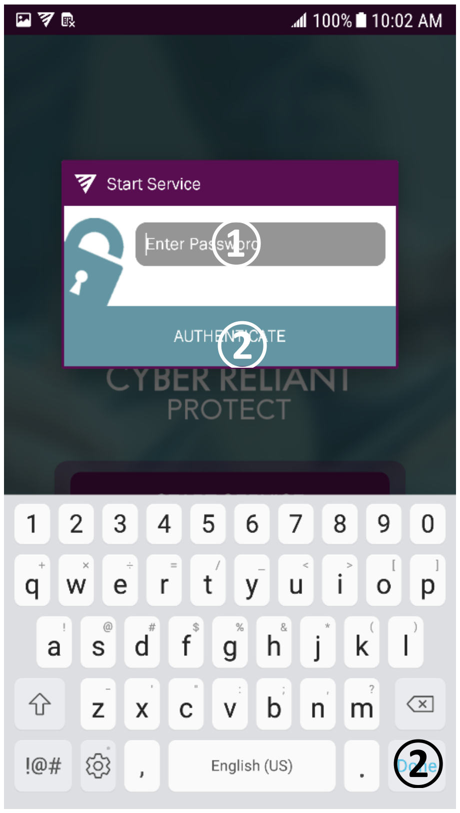


Figure 5 Authentication

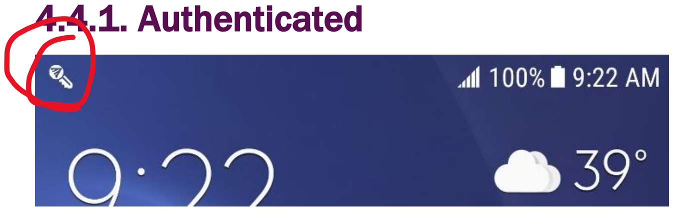
Field	Description	Input Type
1. Password	Input User password	text
2. Submit	Submit the password for Authentication.	Button

Once Authenticated any Integrated Application, in this case the **CRC Data IO Protection Plugin** can start initializing their protection core at startup of the app. If not Authenticated, the Integrated Applications will not startup.

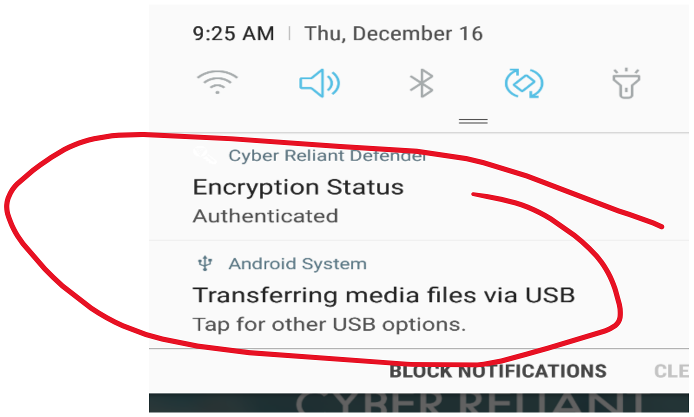
4.4. Status Bar:

Once started the current state of Authentication is displayed by an Icon in the Status Bar.

4.4.1. Authenticated



If the user drops down the status bar, a selection option is offered :


9:25 AM | Thu, December 16

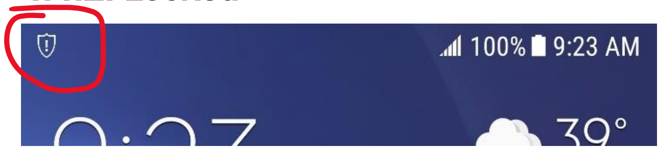
Cyber Reliant Defender
Encryption Status
Authenticated

Android System
Transferring media files via USB
Tap for other USB options.

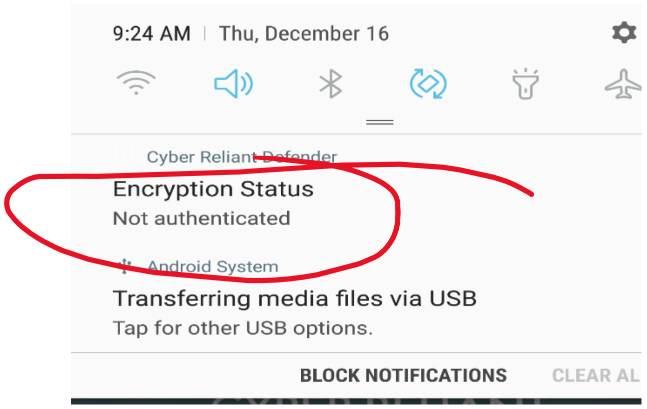
BLOCK NOTIFICATIONS CLE

And when selected displays the Authenticated Utility screen below.

4.4.2. Locked



If the user drops down the status bar:


9:24 AM | Thu, December 16

Cyber Reliant Defender
Encryption Status
Not authenticated

Android System
Transferring media files via USB
Tap for other USB options.

BLOCK NOTIFICATIONS CLEAR AL

When selected it pops up the Authentication Dialog.

4.5. Utility Screen

4.5.1. Authenticated

Once Authenticated for the first time the Utility scree will be displayed. This is the interface that will be displayed at any time the Management Service is selected from the App interface.

Utility functionality:

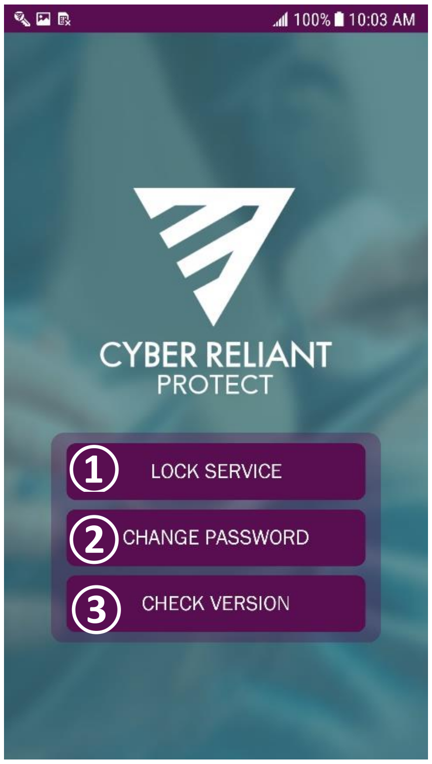


Figure 6 Utility Screen (Authenticated)

1. LockService	De-Authenticate the Device. No Integrated Applications will be able to encrypt until Authenticated. Once the service is locked, this selection changes to ‘UNLOCK SERVICE’ which will bring up the Authentication screen for user password input.	Button
2. Change Password	Allows the user to change password as above.	Button
3. Check Version	Performs a version check of Cyber Reliant Protect to determine if the installed version is up-to-date. Internet connection required.	Button

4.5.2. Unauthenticated

If the service has not yet been authenticated, or the service has been ‘Locked’ The utility screen functionality is slightly different:

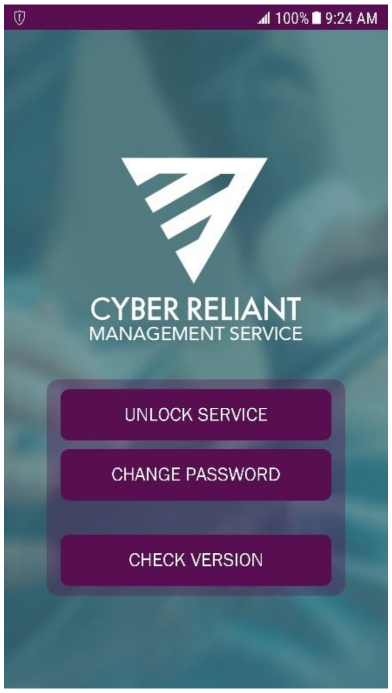


Figure 7 Utility Screen (Locked)

1. UnLockService	Call the password dialog to Authenticate the Device. No Integrated Applications will be able to encrypt until Authenticated.	Button
2. Change Password	Allows the user to change password as above.	Button
3. Check Version	Performs a version check of Cyber Reliant Protect to determine if the installed version is up-to-date. Internet connection required.	Button

4.6. Default Configuration:

To streamline the testing process, the management service was embedded with a default configuration. The values selected for the configuration were chosen as the most appropriate values for testing and normal distribution.

Below are the Configuration settings embedded in the current Management Service release:

Field	Description	Pre-set value
1. Apps to Encrypt	Launches a popup for the user to select which applications installed on the EUD will be authorized to handle encrypted data	CRC.FileUtility.filebrowser
2. Select M:N	Set how many shreds each encrypted file will be split into (N), and how many of those shreds are required to recombine the file (M). Options include: 1:1, 2:2, 2:3, 3:3, 2:4, 3:4, 4:4, 3:5, 4:5, 5:5	2:2
3. Server Address	IP address of the TCM (if applicable)	N/A

Field	Description	Pre-set value
4. Password Requirements	Sets requirements for the complexity of the user password. Options include: Simple: at least 6 characters (any characters). Medium: at least 8 characters, must have at least one lowercase, uppercase, number, and a special character. Complex: at least 16 characters, must have at least two lowercase, uppercase, number, and a special character.	Simple
5. Authentication Mechanism	Sets how the user will use their password to authenticate. "Device" requires authentication every time the device is unlocked. "Timer" requires authentication after a certain duration of time. "Boot" requires authentication every time the device is turned on.	Boot
6. Authentication Timeout	Sets how long the device remains authenticated before requiring re-authentication. Setting is hidden unless Authentication Mechanism is set to "Timer".	N/A
7. Maximum Attempts	Sets the maximum number of consecutive incorrect password attempts that are allowed prior to user lock out.	Max (10)
8. Lockout Time	Sets the length of time that the device remains locked after the maximum amount of password attempts have been made.	1 minute
9. New Password	Password used for authentication. Must meet the Password Requirements.	ATAKPW
10. Confirm New Password	Re-entry of password used for authentication. String must match the New Password textbox.	ATAKPW

4.7. Permissions Used in Installation

The Management Service uses the following Android permissions:

Permissions	Explanation
android.permission.READ_PHONE_STATE	Phone status and identity access is requested so that the phone's ID can be used for identification purposes.
android.permission.QUERY_ALL_PACKAGES	This permission is requested to enable the management service to monitor application install/uninstall for third party app registration/removal.
android.permission.ACCESS_WIFI_STATE android.permission.CHANGE_WIFI_STATE	The ability to modify the Wi-Fi connection is requested so that the device's MAC address can be used for identification purposes. The MAC address may only be available from

	one of a few sources, so multiple Wi-Fi-related permissions are requested to make sure all of these sources are available if needed.
android.permission.INTERNET android.permission.ACCESS_NETWORK_STATE	Network access is requested for making network requests to the Trivalent licensing server, using the HTTPS.
android.permission.BLUETOOTH	Bluetooth permission is requested so that the device's Bluetooth adapter's name and address can be used for identification.
android.permission.REORDER_TASKS	The ability to reorder apps is used because the Management Service needs the ability to launch and display authentication prompts from the background.
android.permission.RECEIVE_BOOT_COMPLETED	The ability to run at startup is used so that the Management Service's authentication service can begin running on Boot.
android.permission.WAKE_LOCK	The Management Service uses this permission because time-based authentication methods need to remain accurate in their de-authentication triggers, even if a device's screen is off.
android.permission.BIND_DEVICE_ADMIN	Necessary for implementing a device administration component.
android.permission.FOREGROUND_SERVICE	Needed for Authentication task as well as status bar notifications
android.permission.WRITE_EXTERNAL_STORAGE	Needed to write externally accessible Logs in future versions.
android.permission.AURICFS_ADMIN	Added for future updates.

4.8. Android Security API Functions Used

4.8.1. Android KeyStore-based RSA Key Pair Generation Encruption/Decryption

Constructors Called

- android.security.KeyPairGeneratorSpec.Builder.ctor(Context)
- javax.security.auth.x500.X500Principal.ctor(String)

Methods Called

- java.security.KeyStore.getInstance(String)
- java.security.KeyStore.load(InputStream, char[])
- java.security.KeyStore.getEntry(String, KeyStore.ProtectionParameter);
- java.security.KeyPairGenerator.getInstance(String, String)
- android.security.KeyPairGeneratorSpec.Builder.setAlias(String)
- android.security.KeyPairGeneratorSpec.Builder.setStartDate(Date)
- android.security.KeyPairGeneratorSpec.Builder.setEndDate(Date)
- android.security.KeyPairGeneratorSpec.Builder.setSerialNumber(BigInteger)
- android.security.KeyPairGeneratorSpec.Builder.setSubject(X500Principal)
- android.security.KeyPairGeneratorSpec.Builder.build()
- java.security.KeyPairGenerator.initialize(AlgorithmParameterSpec)
- java.security.KeyPairGenerator.generateKeyPair()
- java.security.KeyStore.PrivateKeyEntry.getCertificate()
- java.security.Certificate.getPublicKey()
- java.security.KeyStore.PrivateKeyEntry.getPrivateKey()
- javax.crypto.Cipher.getInstance(RSAAlgorithm);
- javax.crypto.Cipher.init(Cipher.***ENCRYPT_MODE***, SecretKey);
- javax.crypto.Cipher.init(Cipher.***DECRYPT_MODE***, SecretKey);
- javax.crypto.Cipher.doFinal()

Objects Created

- java.security.KeyStore
- java.security.KeyStore.PrivateKeyEntry
- java.security.KeyStore.Entry
- java.security.KeyPairGenerator
- android.security.KeyPairGeneratorSpec
- android.security.KeyPairGeneratorSpec.Builder
- javax.security.auth.x500.X500Principal
- java.security.KeyPair
- java.security.PublicKey
- java.security.PrivateKey
- java.security.cert.Certificate
- javax.crypto.Cipher

Exceptions Handled

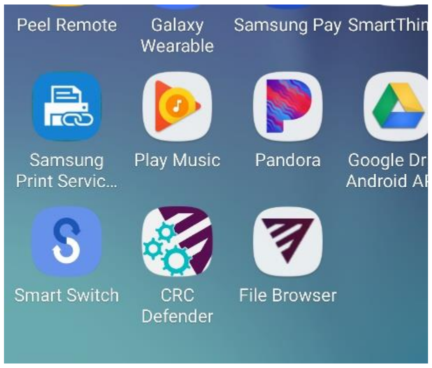
- java.security.KeyStoreException
- java.security.Cert.CertificateException
- java.security.NoSuchAlgorithmException
- java.security.UnrecoverableEntryException
- java.security.NoSuchProviderException
- java.security.InvalidAlgorithmParameterException

5. File Browser App:

The File Browser application is a utility to allow Encryption/ Decryption (Protection) of user selected files and directories of files. This only includes access to publicly accessible files once the Management Service is Started and Authenticated.

In this case, the Integrated Application is '**FileBrowser**' and it is registered with the Management Service using the 'CRC.FileUtility.filebrowser' app ID.

As with all integrated and registered 3rd party applications the File Browsers ability to protect files is controlled by the Authentication status of the Management Service (Defender).



5.1. Startup

Once installed on startup of the application the app reaches out to the Management service to get authentication status. If the Authentication is 'Locked' the app starts up with a blank scree and offers no functionality. If the system is authenticated then the app opens to a file browse and selection screen:

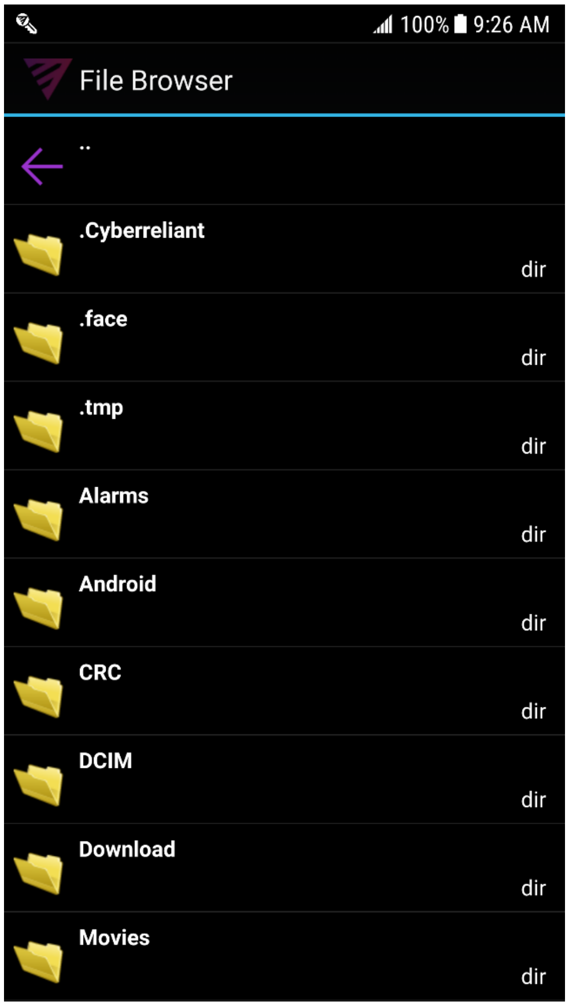


Figure 8 Startup Screen

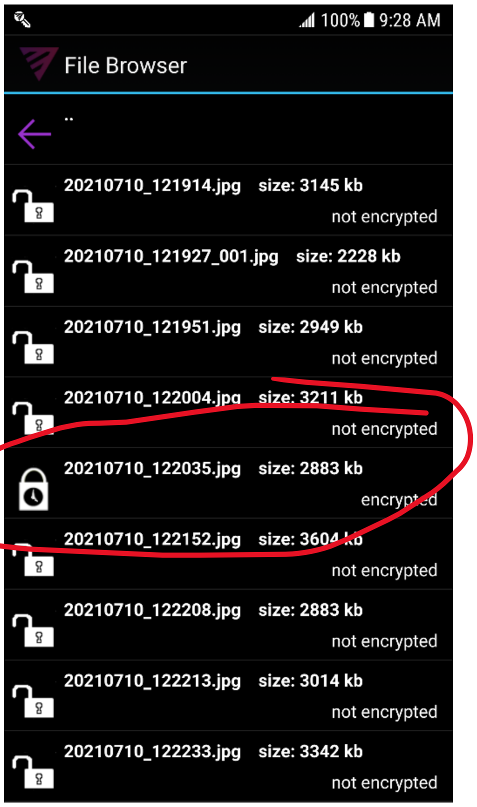
This is the main selection screen. Single press selection of any directory entry will open the screen to the selected directory content. Long-press selection will offer the choice of encrypting/decrypting the entire directory content of files as well as being able to select performing the operation on all subdirectories (recursive).

A single press selection on a single file will offer the user the option to encrypt or decrypt the file depending on its current state.

The ‘Back Arrow’ At the top of the directory list allows moving back out of the directory to the parent directory.

5.2. Single File Selection

Once a directory is selected it will show its content. If there are files contained in the directory, they will be displayed with their encryption status as well as file size:


100% 9:28 AM

File Browser

..

20210710_121914.jpg size: 3145 kb not encrypted

20210710_121927_001.jpg size: 2228 kb not encrypted

20210710_121951.jpg size: 2949 kb not encrypted

20210710_122004.jpg size: 3211 kb not encrypted

20210710_122035.jpg size: 2883 kb encrypted

20210710_122152.jpg size: 3604 kb not encrypted

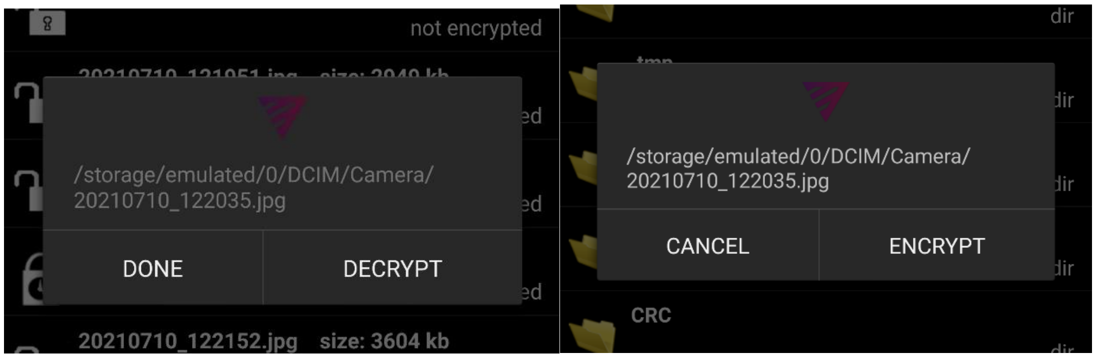
20210710_122208.jpg size: 2883 kb not encrypted

20210710_122213.jpg size: 3014 kb not encrypted

20210710_122233.jpg size: 3342 kb not encrypted

Figure 9 File List Display

By selecting any single file by either a short or long press the appropriate encryption dialog will be displayed:


/storage/emulated/0/DCIM/Camera/
20210710_122035.jpg

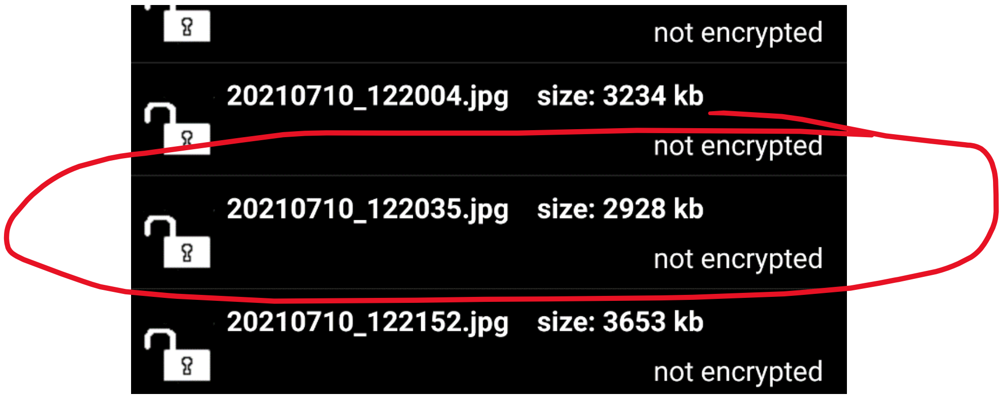
DONE DECRYPT

/storage/emulated/0/DCIM/Camera/
20210710_122035.jpg

CANCEL ENCRYPT

Figure 10 Encrypt/Decrypt Dialog Selection

Selecting the ‘Encrypt’ or ‘Decrypt’ button will perform the selected operation on the file and the file directory will be updated to show the files new encryption state:



not encrypted

20210710_122004.jpg size: 3234 kb
not encrypted

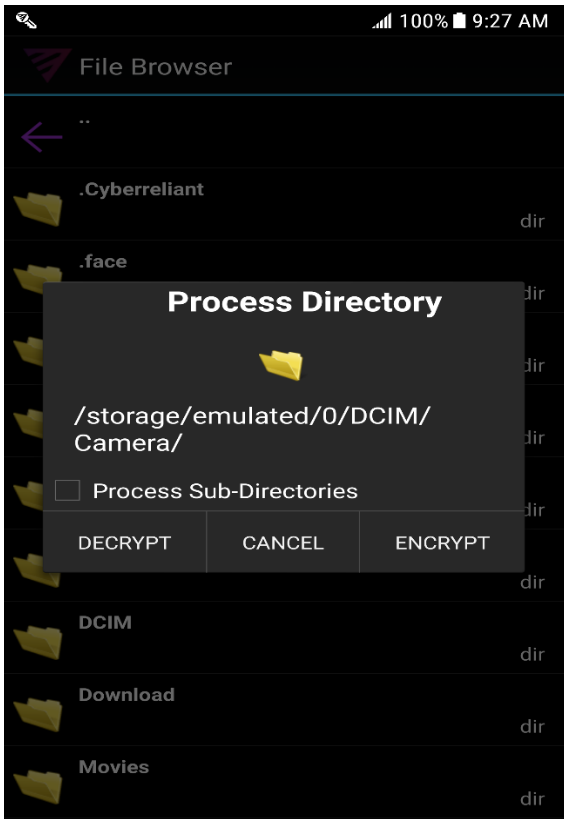
20210710_122035.jpg size: 2928 kb
not encrypted

20210710_122152.jpg size: 3653 kb
not encrypted

Figure 11Updated File State

5.3. Directory Selection

In the case that the user wants to Encrypt/Decrypt an entire Directory of files, this is made possible by the user selecting the Directory with a 'Long-press' selection.



The Process Directory Dialog is used to select what the user wants to do with the directory of files.

5.3.1. Decrypt

If decrypt is chosen, then all encrypted files in the directory will be decrypted and all others will be untouched.

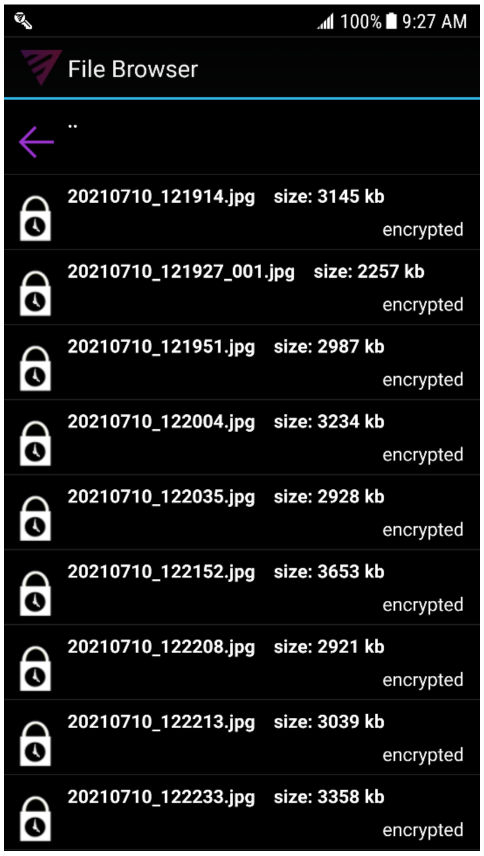
5.3.2. Encrypt

If Encrypt is chosen then all un-encrypted files in the directory will be encrypted.

5.3.3. Process Sub-Directories

If the selection is checked, the same action taken on the selected directory will be used on each of the directories subdirectories (recursive). If un-checked it will ignore all sub-directories in its processing.

As with the single file processing, once processed the UI will be updated to reflect the current file state.



20210710_121914.jpg size: 3145 kb
encrypted
20210710_121927_001.jpg size: 2257 kb
encrypted
20210710_121951.jpg size: 2987 kb
encrypted
20210710_122004.jpg size: 3234 kb
encrypted
20210710_122035.jpg size: 2928 kb
encrypted
20210710_122152.jpg size: 3653 kb
encrypted
20210710_122208.jpg size: 2921 kb
encrypted
20210710_122213.jpg size: 3039 kb
encrypted
20210710_122233.jpg size: 3358 kb
encrypted

5.4. Permissions Used in Installation

The File Browser app uses the following Android permissions:

Permissions	Explanation

android.permission.MANAGE_EXTERNAL_STORAGE	Allows use of the ACTION_MANAGE_ALL_FILES_ACCESS_PERMISSION intent action to direct users to a system settings page where they can enable the following option for your app: Allow access to manage all files.
android.permission.WRITE_EXTERNAL_STORAGE android.permission.READ_EXTERNAL_STORAGE	Enable application to read and write files into the device shared drives.
android.permission.QUERY_ALL_PACKAGES android.permission.USE_CREDENTIALS	Allows communication with Management Service.
android.support.PARENT_ACTIVITY	Added make it easy for users to find their way back to the app's main screen on older Android versions.

5.5. Android Security API Functions Used

5.5.1. Android KeyStore-based RSA Key Pair Generation Encruption/Decryption

Constructors Called

- android.security.KeyPairGeneratorSpec.Builder.ctor(Context)
- javax.security.auth.x500.X500Principal.ctor(String)

Methods Called

- java.security.KeyStore.getInstance(String)
- java.security.KeyStore.load(InputStream, char[])
- java.security.KeyStore.getEntry(String, KeyStore.ProtectionParameter);
- java.security.KeyPairGenerator.getInstance(String, String)
- android.security.KeyPairGeneratorSpec.Builder.setAlias(String)
- android.security.KeyPairGeneratorSpec.Builder.setStartDate(Date)
- android.security.KeyPairGeneratorSpec.Builder.setEndDate(Date)
- android.security.KeyPairGeneratorSpec.Builder.setSerialNumber(BigInteger)
- android.security.KeyPairGeneratorSpec.Builder.setSubject(X500Principal)
- android.security.KeyPairGeneratorSpec.Builder.build()
- java.security.KeyPairGenerator.initialize(AlgorithmParameterSpec)
- java.security.KeyPairGenerator.generateKeyPair()

- java.security.KeyStore.PrivateKeyEntry.getCertificate()
- java.security.Certificate.getPublicKey()
- java.security.KeyStore.PrivateKeyEntry.getPrivateKey()
- javax.crypto.Cipher.getInstance(RSAAlgorithm);
- javax.crypto.Cipher.init(Cipher.***ENCRYPT_MODE***, SecretKey);
- javax.crypto.Cipher.init(Cipher.***DECRYPT_MODE***, SecretKey);
- javax.crypto.Cipher.doFinal()

Objects Created

- java.security.KeyStore
- java.security.KeyStore.PrivateKeyEntry
- java.security.KeyStore.Entry
- java.security.KeyPairGenerator
- android.security.KeyPairGeneratorSpec
- android.security.KeyPairGeneratorSpec.Builder
- javax.security.auth.x500.X500Principal
- java.security.KeyPair
- java.security.PublicKey
- java.security.PrivateKey
- java.security.cert.Certificate
- javax.crypto.Cipher

Exceptions Handled

- java.security.KeyStoreException
- java.security.Cert.CertificateException
- java.security.NoSuchAlgorithmException
- java.security.UnrecoverableEntryException
- java.security.NoSuchProviderException
- java.security.InvalidAlgorithmParameterException