Enabler Manual
Device-Based Anonymization

Project name	5G Enablers for Network and System Security and Resilience
Short name	5G-ENSURE
Grant agreement	671562
Call	H2020-ICT-2014-2
Authors	TIIT: Madalina Baltatu, Luciana Costa, Dario Lombardo

Foreword

5G-ENSURE belongs to the first group of EU-funded projects which collaboratively develop 5G under the umbrella of the 5G Infrastructure Public Private Partnership (5G-PPP) in the Horizon 2020 Programme. The overall goal of 5G-ENSURE is to deliver strategic impact across technology and business enablement, standardization and vision for a secure, resilient and viable 5G network. The project covers research and innovation - from technical solutions (5G security architecture and testbed with 5G security enablers) to market validation and stakeholders engagement - spanning various application domains.

The present document is the manual of the Device-based Anonymization enabler from WP3 Task T3.2 – Privacy, and is part of deliverable D3.8 of the 5G-ENSURE project. It provides the installation and administration guide, the user and programmer guide and the tests to be performed in order to correctly install, configure and use the enabler. An example deployment use case is also illustrated in order to facilitate users' understanding of the provided features.

Disclaimer

Copyright notice

Contents

1 Introduction

This enabler provides anonymization techniques on the user's device, offering protection against disclosure of the IMSI stored on the SIM. The privacy/anonymization configuration (or profile) is directly controlled by the user, who can enable different anonymization profiles. The detailed specification can be found in deliverables D3.5 [1] and especially D3.6 [2].

The OS running on the device implements a specific IMSI anonymization algorithm at the lowest possible layer in the device OS stack, and offers to the user the means to switch on and off the anonymization. As illustrated in Figure 1, whenever a user space application requires access to the IMSI stored on the SIM and protected by an active privacy profile/configuration, the request will be managed by a privacy provider, which will return an anonymized version of the IMSI to the caller. The requesting application will obtain the anonymized piece of data instead of the original one.

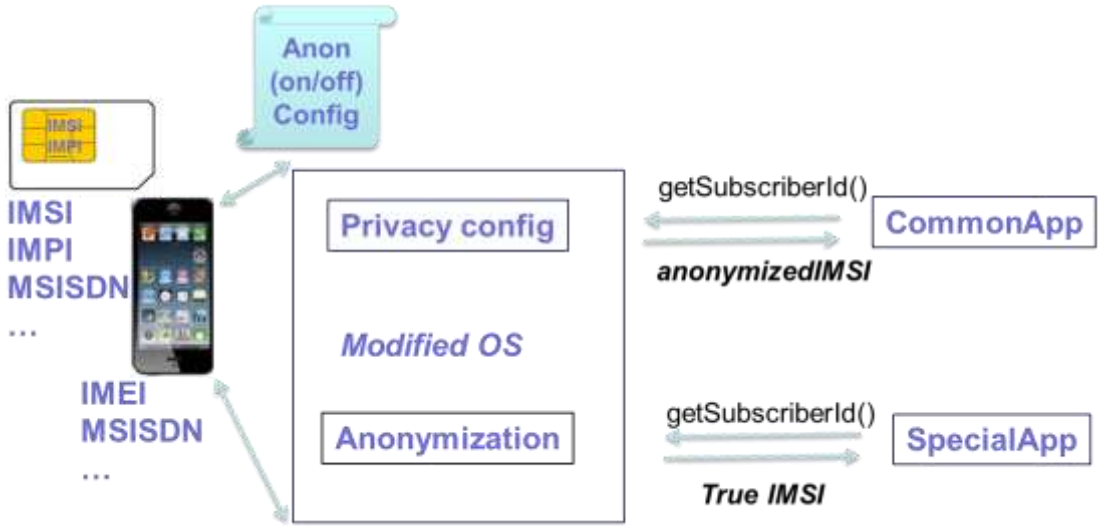

Figure 1 Device-based anonymization - high level architecture.

At a high level, the enabler implements two basic components (features):

- A format preserving anonymization algorithm which provides the anonymization function for all data received in input (i.e., the IMSI), with the preservation of the input data format. The algorithm is embedded in the operating system code, as low as possible in the kernel stack.
- Privacy Configuration: the mediator between the user and the anonymizing function of the device, which gives the user a means to enable the anonymization capabilities to protect sensitive data and avoid their disclosure to user space applications. Such anonymization capabilities can be further customized by white-listing single applications and by allowing them to access the sensitive data in clear text.

2 Installation and Administration Guide

The enabler is an Android OS image build from the CyanogenMod 13.1 [3] milestone sources modified to include all the features that provide IMSI anonymization and the capability to configure it.

2.1 System Requirements

In order to run the enabler, the user will only need a Samsung Galaxy SII device. The enabler is based on a custom release of the Android operating system build from the CyanogenMod 13.1 milestone sources for the Samsung Galaxy SII device and provides a system-wide device-based data anonymization which supports customized per-application security profiles.

The method returning the IMSI to the user space applications is intercepted in the TelephonyManager system service component of the Android platform [4] and the returned value is encrypted using a format-preserving algorithm [5]. The encryption is based on a secret key generated using AES-256 that is refreshed at every device boot.

2.2 Enabler Configuration

In the default post-installation enabler configuration, both the system-wide and per-application data anonymization features are set to off.

In order to enable either of these options the user needs to navigate through the graphical interface and set the desired configuration by accessing the Security menu from the device Settings panel.

2.3 Enabler Installation

The current version of the enabler is built from the source code of CyanogenMod 13.0 [3] for the Samsung Galaxy SII [6] and is based on a Java implementation of the FPE algorithm [5].

CyanogenMod is an open-source operating system based on AOSP (Android Open Source Project) [7] that offers a wide range of features such as unlockable bootloader, root access, layout and graphical customization and tweaks, CPU overclocking and other performance enhancements. The CM SDK has been used to customize the Security activity in the Settings.apk and store the device anonymization toggle state.

An existing Java implementation of the FPE algorithm has been modified and added to the CyanogenMod 13.0 code base in the form of an external library in order to grant encryption functionalities in the TelephonyManager system service.

Before flashing the CyanogenMod-based ROM image that can be downloaded from the project's Artifact repository, it is required to install a custom recovery to the recovery partition of the device and to root it.

2.3.1 Installing a custom recovery software

A custom recovery software is needed in order to flash any custom ROM image. The open-source community project TWRP has been used for our enabler. The correct image can be found in [4]. TWRP can be installed by using both Linux and Windows based machines but we have had the most success by using Windows-based systems (there were potential compatibility issues on USB Samsung drivers, flashing software in our tests on Ubuntu).

There are several requirements that need to be addressed before proceeding to the installation of TWRP using Windows:

- Download and install Samsung USB drivers (http://developer.samsung.com/galaxy/others/android-usb-driver-for-windows)
- Download the latest TWRP software for the Samsung Galaxy S2 (https://eu.dl.twrp.me/i9100/)

- Download ODIN 3.10.7 (http://www.cyanogenmods.org/forums/topic/install-twrp-recovery-samsung-android-using-odin/)
- Download and install the SDK platform tools (https://developer.android.com/studio/releases/platform-tools.html) that include adb.exe and fastboot.exe
- Enable Developer Options by navigating to Settings -> About Phone and tapping on Build number 7 times
- Enable Android debugging in Settings -> Developer Options

At this stage the TWRP install procedure can be started:

- Turn off the device or restart the device
- Put the device in download mode by pressing and holding the volume-down button, the power button and the home button while the device is booting up. At the warning sign screen press the volume-up button to enter the download mode
- When the device is in download mode, connect it to the Windows machine using a USB cable
- Open ODIN. ODIN will display a "<ID:0/00X> Added!!" message if the device is correctly detected
- Press the AP button in the ODIN graphical interface and find the .tar file of TWRP
- Make sure that no options are checked
- Press the Start button

2.3.2 Rooting the device

This installation stage requires the user to download the latest recovery flashable SuperSU .zip file (http://www.supersu.com/download).

The SuperSU .zip file can be flashed by sideloading it using adb:

- Turn off the device
- Enter TWRP by booting it while holding the volume-up button, the power button and the home button
- In TWRP navigate to Advanced -> ADB Sideload
- On the installation machine, open a command prompt and execute "adb sideload <path_to_supersu_flashable_zip_file>"

2.3.3 Flashing the enabler's image

After downloading the flashable zip file from the project's artifacts repositories:

- Turn off the device
- Enter TWRP by booting it while holding the volume-up button, the power button and the home button
- In TWRP navigate to Advanced -> ADB Sideload
- On the installation machine, open a command prompt and execute "adb sideload <path_to_enabler_flashable_zip_file>"

2.4 Troubleshooting

This section includes the problems we encountered during the enabler installation (flashing). For error conditions in the installation of the custom recovery software and rooting tool please refer to their corresponding web pages.

2.4.1 Device bootloops after flashing ROM

If the device bootloops after flashing the custom ROM, it is advisable to attempt repairing steps in this order:

- From TWRP wipe the Dalvik cache and the device cache from Wipe -> Advanced wipe
- From TWRP perform a factory reset from the Wipe menu
- Restore the original ROM and attempt the whole install procedure again.

3 User and Programmer Guide

This subsection describes how to use the enabler after the enabler's image is flashed on the device. The enabler is not programmable.

3.1 User Guide

System-wide anonymization can be toggled on or off by navigating to Settings → Security. When enabled, installed applications will only be able to access encrypted IMSI since all IMSI requests will be intercepted by the enhanced privacy module.

In Settings → Security → Privacy Settings it is possible configure IMSI access on a per-application basis. This feature can be used to whitelist specific applications while leaving system-wide anonymization enabled.

3.2 Programmer Guide

The enabler is not programmable.

4 Unit Tests

This subsection describes some simple tests for the enabler (and its components) in order to check that the IMSI anonymization and the configuration of anonymization both work as expected.

4.1 Information about Tests

The IMSI anonymization privacy feature is enabled system-wide and the device come with 2 applications, it.tim.goodware and it.tim.malware, that are used to test the anonymization and configuration features.

In the test scenario, the user has to specify a privacy profile that allows one of the two applications (it.tim.goodware) to access the real IMSI.

The privacy profile that is set for the second application (it.tim.malware) forbids obtaining the real IMSI and, therefore, this app will get an anonymized IMSI, i.e., the IMSI encrypted by means of a format preserving encryption algorithm.

4.2 Unit Test 1

The users opens Settings and check the privacy profile of the app it.tim.goodware.

The user launches the it.tim.goodware application and tries to retrieve the IMSI by tapping the “GET IMSI” button. The operating system returns the real IMSI to the application that displays this value to the user.

Figure 2 Goodware app activity.

4.3 Unit Test 2

The users opens Settings and check the privacy profile of the app it.tim.malware.

The user then executes the it.tim.malware application and taps the “GET IMSI” button. In this case the user observes that the application is not allowed to fetch the real IMSI. The displayed IMSI has in fact been encrypted using a format-preserving encryption and this anonymized version of IMSI is returned to the app instead of the real one.

Figure 3 Malware app activity.

5 Abbreviations

APK	Android application package is an archive file format used to distribute and install mobile applications and middleware.
AOSP	Android Open Source Project
IMSI (MCC, MNC, MSIN)	The International Mobile Subscriber Identity is a unique number that identifies the user of a cellular network. An IMSI is usually 15 digits in length. The first 3 digits represent the MCC (mobile country code) and are followed by the MNC (mobile network code) which is either 2 digits (European standard) or 3 digits (North American standard). The remaining digits are the MSIN (Mobile Subscription Identification Number), that is a unique identifier within the network's customer base.
FPE	Format Preserving Encryption
FFX	FFX is a mode of operation for format-preserving encryption (FPE). FPE algorithms work in such a way that the cipher text is represented in the same format as the input. In the device-based anonymization the format of the input and of the cipher text refers to the number of digits.
5G-PPP	5G Infrastructure Public Private Partnership
ADB	Android Debug Bridge

References

[1] 5G-Ensure Deliverables, "Deliverable D3.5, 5G-ENSURE_D3.5 5G-PPP security enablers technical roadmap (Update)," 2017. [Online]. Available: https://5gensure.eu/deliverables.

[2] 5G-Ensure Deliverables, "Deliverable D3.6, 5G-ENSURE_D3.6 5G-PPP security enablers open specifications (v2.0)," 2017. [Online]. Available: https://5gensure.eu/deliverables.

[3] Cyanogen, "Cyanogenmod 13.0," [Online]. Available: https://github.com/CyanogenMod.

[4] Google, "Android Platform Architecture," [Online]. Available:

https://developer.android.com/guide/platform/index.html.

[5] robshep, "FPE for Java," [Online]. Available: https://github.com/robshep/JavaFPE.

[6] Samsung, "Samsung Galaxy S2 i9100," [Online]. Available: https://twrp.me/devices/samsunggalaxys2i9100.html.

[7] Google, "Android Open Source Project," [Online]. Available: https://source.android.com/.