National Science Foundation CNS 10-16193

User Guide for iTrust over SMS and iTrust over Wi-Fi Direct Installation and Use

University of California, Santa Barbara

by Isaí Michel Lombera

This documentation explains how to install iTrust over SMS, to integrate iTrust over Wi-Fi Direct, and to search for and retrieve information on Android mobile devices.

Table of Contents

Overview

The iTrust system is a decentralized information publication, search and retrieval system designed to enable users to share information over the Internet, cellular telephony networks, and mobile ad-hoc networks using HTTP, SMS, and Wi-Fi Direct, respectively.

This documentation details how to use iTrust over SMS within cellular telephony networks and how to use iTrust over Wi-Fi Direct within mobile ad-hoc networks. For iTrust over SMS, we show how to install the app on an Android mobile phone platform and how to use the graphical user interface to distribute and search for information. For iTrust over Wi-Fi Direct, we show how to download the code and integrate it into your existing app source code on the Android mobile phone platform.

iTrust over SMS

Getting the iTrust over SMS app onto your mobile phone is relatively easy and straightforward. Once installed on an appropriate mobile device, using iTrust over SMS is also relatively easy.

Requirements

First, you must have an Android mobile phone with an SMS text messaging service plan or contract. You are responsible for all costs incurred by sending and receiving SMS messages; iTrust over SMS sends/receives on average four messages for each query and match, so plan accordingly. We recommend an unlimited SMS service plan because such plans are relatively inexpensive. iTrust over SMS was tested on Android version 2.1 and above, so we recommend that you use at least Android version 2.1.

By default, Android does not allow third party apps (including user-created apps) to be installed on a mobile phone. Enable third party app installation by navigating to Apps->Settings->Security. Then tap 'Unknown Sources' to enable third party apps (see Figure 1).

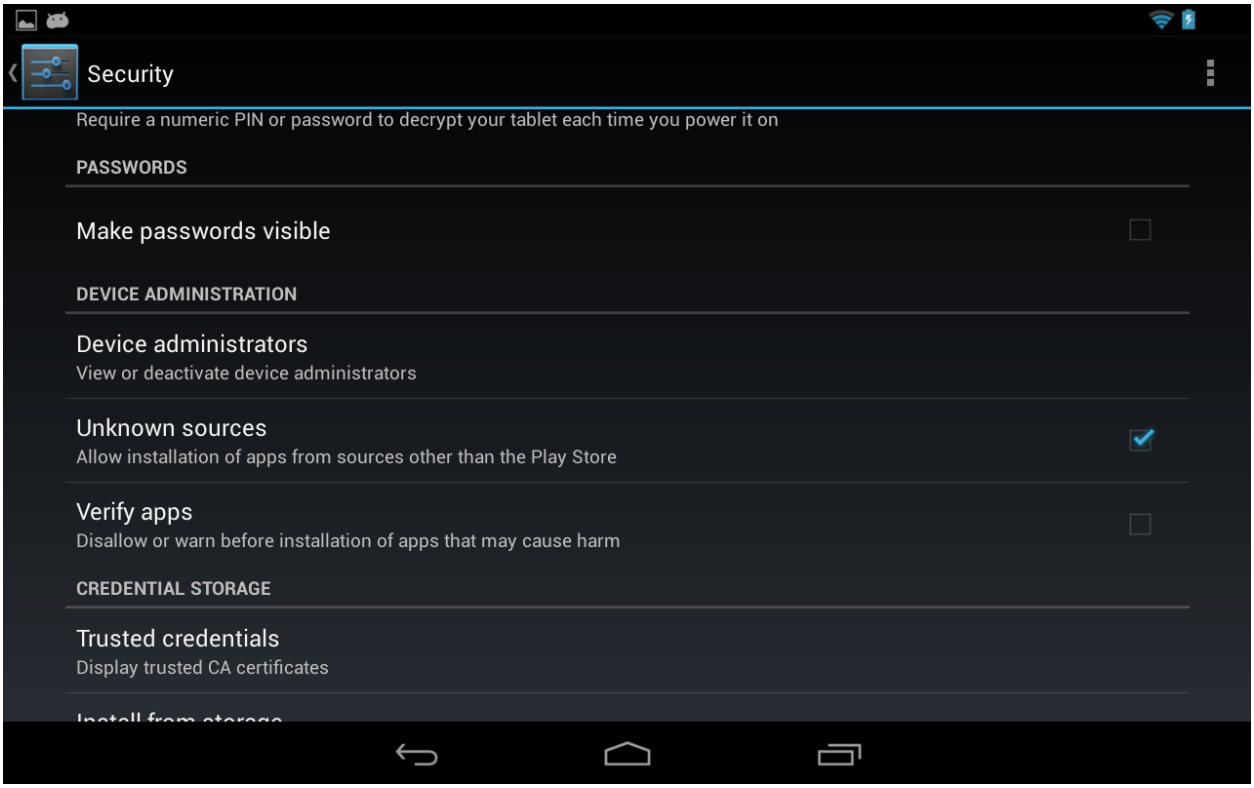

Figure 1. Enabling third party app downloading on a Nexus 7 tablet.

Installation Walk-Through

The basic process for installing iTrust over SMS on a mobile device is to transfer the app (APK) file onto the device and then to begin the installation procedure. Doing so can be accomplished most easily by opening the iTrust Web site on the Internet browser of the mobile device, downloading the app, and tapping the file to initiate the automated installation procedure.

Startup the Browser app on your device (by default the Google Chrome browser) and go to:

http://itrust.ece.ucsb.edu

On the Web page menu, under the header image, tap the Code menu item, which will take you to the source code download section. Then, tap the iTrust over SMS Android app link to begin downloading the app.

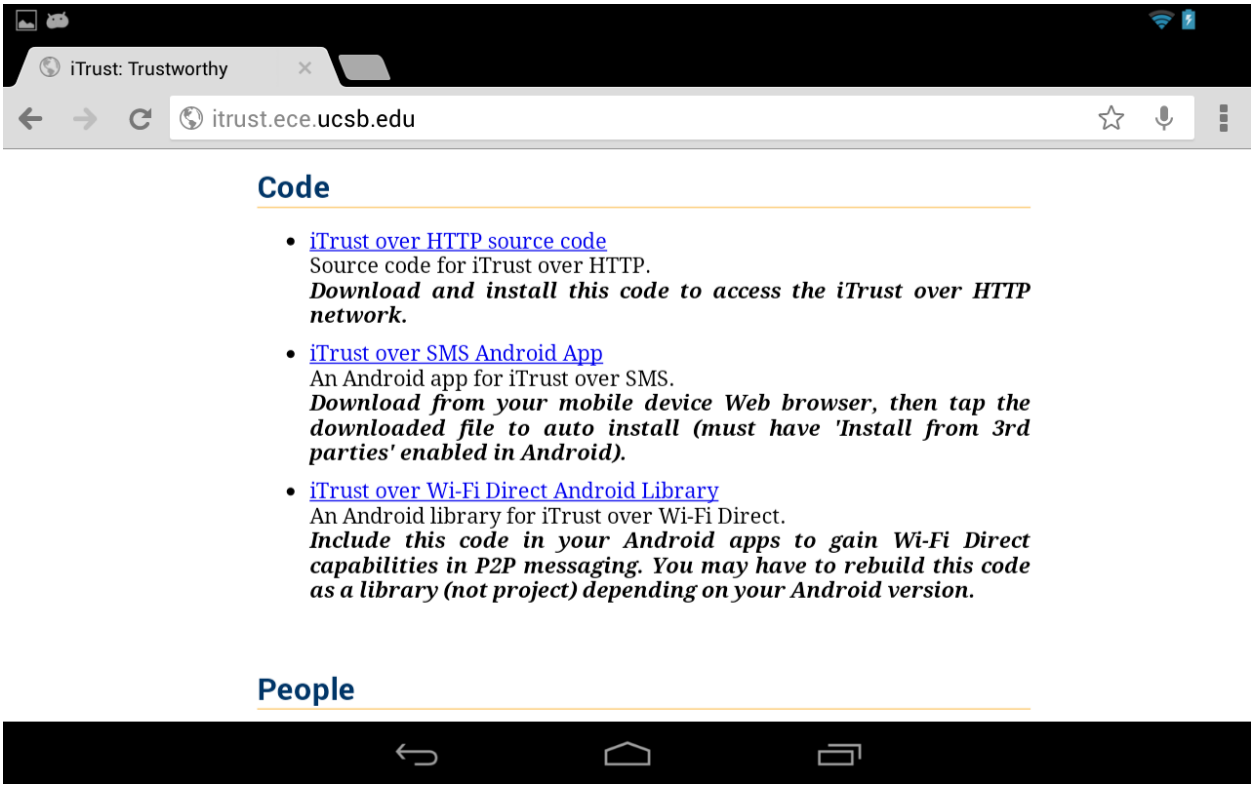

Figure 2. Viewing the iTrust Web site for file downloading on a Nexus 7 tablet.

You will see the download progress notification in the upper right corner. When the download is complete, tap the notification bar and then tap the file that you just downloaded. The iTrust over SMS app will automatically install itself.

Using the Graphical User Interface

Start the iTrust over SMS app by tapping Apps->iTrust over SMS.

The first screen you see shows a list of searches/queries made from your device and sent to the iTrust over SMS network (Figure 3). When you make subsequent searches, this list will be updated with your search history automatically. At any time you can tap a query to see its details and perhaps download search hits. Near the bottom you will see a New Search button, tap it to start a new search.

Figure 3. Search list on a Galaxy Nexus.

The new search screen enables you to enter a search query and then to tap the Search now button to have your query distributed across the iTrust over SMS network (Figure 4). After tapping the Search now button, you will be taken back to the first screen which shows your search history.

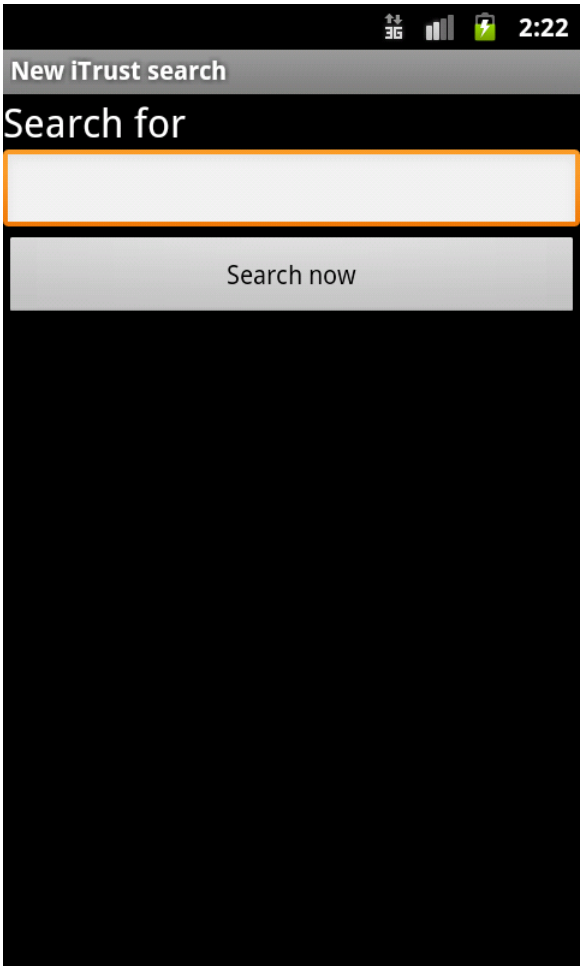

Figure 4. New search on Galaxy Nexus.

You can now tap on any search item to see the Search details screen which includes information such as the search date and the number of nodes to which the query was relayed (Figure 5). If there are search hits, they will be listed on this screen, and you can tap them to fetch the information automatically. The information will then be displayed on the screen by Android using whatever the default app is for handling that type of information. In most cases, the information is text and will simply be displayed on the screen; otherwise, Android will display it in some other way (e.g., pictures are shown in the Gallery app). When you are done viewing the search results, tap the Back button on your phone to go back to the search history screen.

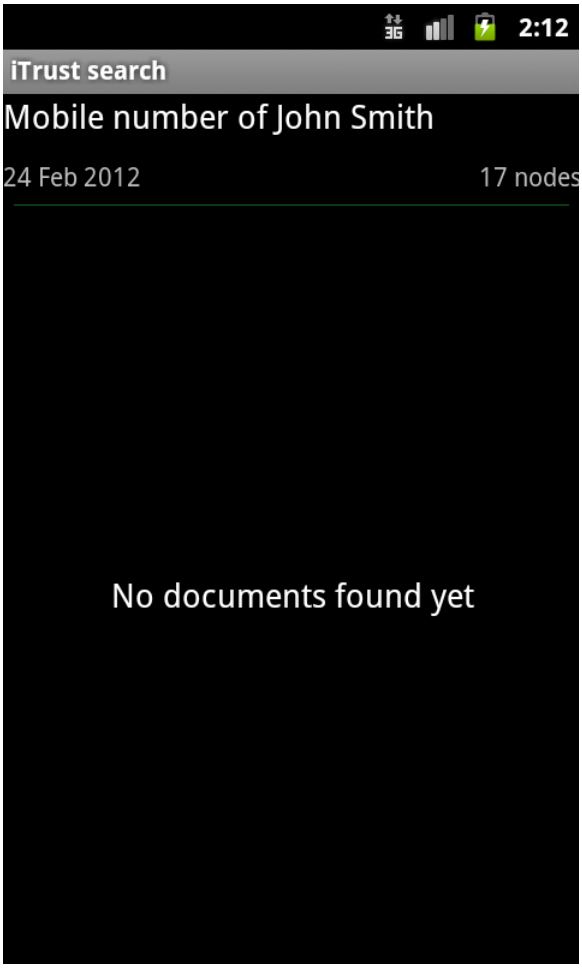

Figure 5. Search details on Galaxy Nexus.

You can alter the default behavior of iTrust over SMS on your device by adding nodes manually and changing the app preferences. Tap the Menu button on your phone to see the Add new node button and Preferences button which take you to the appropriate screens.

At the add new node screen, you can add nodes by tapping the button on the bottom labeled Add new node (Figure 6). Type in the telephone number of the node you want to add and iTrust will add it to the iTrust membership.

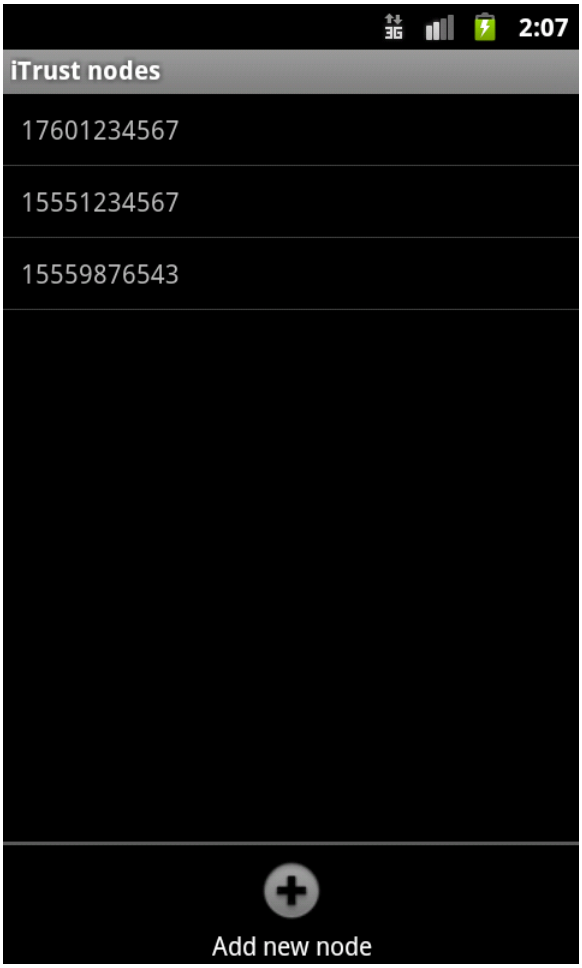

Figure 6. Adding nodes on Galaxy Nexus.

At the preferences screen, you can change a number of settings for the iTrust over SMS app. Each setting has a small explanation beneath it, which describes what the setting does.

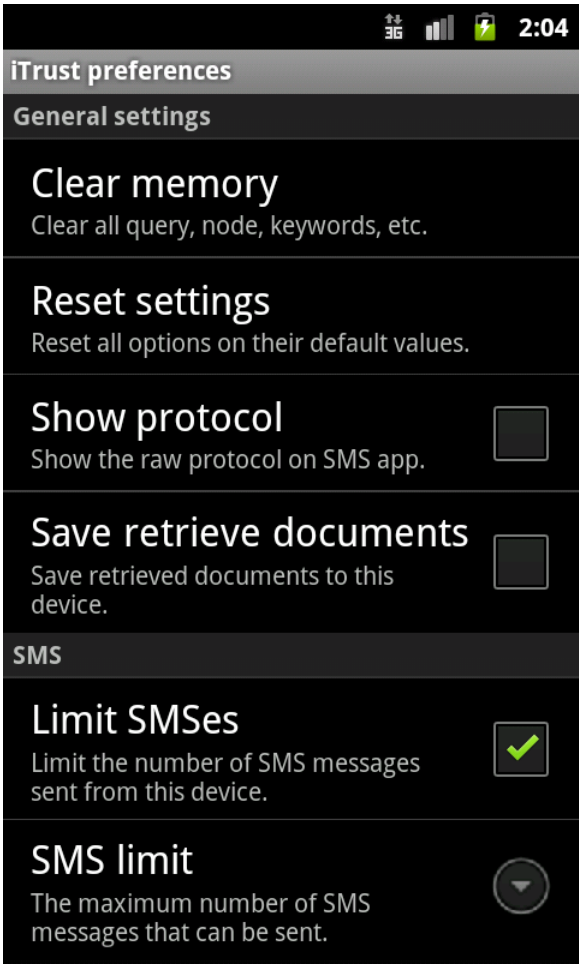

Figure 7. App preferences on Galaxy Nexus.

Finally, note that the app itself does not let you directly manage metadata distribution. It will generate metadata automatically and will distribute the metadata to other nodes in the iTrust membership periodically. However, you can disable metadata generation in the Preferences screen by disabling the Share contacts setting and Share documents setting.

iTrust over Wi-Fi Direct

You can download the iTrust over Wi-Fi Direct source code and easily integrate it into your code. Doing so enables your apps, on suitable hardware, to have access to the iTrust over Wi-Fi Direct network when other Wi-Fi Direct devices are nearby. Figure 8 shows an example app, similar to the Android iTrust over SMS app, built using the iTrust over Wi-Fi Direct code as a library accessed from a typical Android app.

Figure 8. An example iTrust over Wi-Fi Direct app on a Nexus 7 tablet.

Requirements

First, you must have a Wi-Fi Direct enabled Android mobile device, such as a tablet or mobile phone, to run iTrust over Wi-Fi Direct. The code for iTrust over Wi-Fi Direct is designed to run on Android 2.2 or a later version; most Android 2.2 devices (and later versions) are Wi-Fi Direct compatible.

You must be in range of another Wi-Fi Direct device to share information, and you must have the Wi-Fi Direct feature enabled on your device (by default it is enabled).

You must have an Android development environment setup on your computer. The iTrust over Wi-Fi Direct code is written with the Android SDK using normal command line tools and Android Debug Bridge (ADB); however the normal Eclipse Android development environment will work as well.

Obtaining the Source Code

You can download the iTrust over Wi-Fi Direct source code directly from the iTrust Web site. On your computer, startup your Internet browser and go to:

http://itrust.ece.ucsb.edu

On the Web page menu, under the header image, tap the Code menu item to be taken to the source code download section (Figure 2). Click the iTrust over Wi-Fi Direct Library link to begin downloading the tarball. Save the file to your project directory or to a path referenced by your Android app project.

API Examples

Integrate the iTrust over Wi-Fi Direct source code into your Android project by either placing a copy of the Java source files directly into your Android project or referencing a library during your build process. By default the iTrust over Wi-Fi Direct source code is a normal Android app project so you can easily run the example apps provided (to test for correct iTrust over Wi-Fi Direct functionality). However, if you wish to run the project as a library and reference the API from your Android app (similar to a JAR file), you must change the iTrust over Wi-Fi Direct source code from an Android app project to an Android library project. See the Android Development Guidelines on how to do this (normally, just add the line android.library=true to your project.properties file).

A good example of how to integrate the iTrust over Wi-Fi Direct source code into your Android app can be seen in the WifiP2pV2.java file. This file instantiates the iTrust over Wi-Fi Direct library object, generates various test data, populates itself with the test data, queries other nodes for information automatically, and transfers information between nodes. Each step is controlled with a tappable button, and the data are logged on the system for your debugging.

The best way to learn how to use the source code is to try the example. The source code is commented and easy to understand. Here is the basic sequence of commands to publish, search for, and retrieve information using iTrust over Wi-Fi Direct.

Instantiate the iTrust library using:

```
itrust_library = new iTrustLibrary(this);
```

This statement will setup all necessary Android services, databases, broadcast listeners, and signal parsers for you. You can edit the default settings of the node you just created on the device (in iTrustNode.java); however, there is no need to do so in most cases; the defaults are sufficient for most users.

Start the iTrust over Wi-Fi Direct service using:

```
itrust_library.signal.send_message(“”,””);
```

By sending an empty message to an empty destination, you force Android to startup the iTrust over Wi-Fi Direct service to start listening for incoming queries or data transfers. iTrust will then automatically handle all message routing, relaying, parsing, and Wi-Fi connection information. You do not need to do any networking programming.

You can also send a (non-empty) message to a node here, but in most cases it makes sense to listen for incoming messages before sending messages.

Announce yourself to any nearby nodes using:

```
itrust_library.signal.send_this_mac_address(null);
```

This method call simply lets other iTrust over Wi-Fi Direct nodes nearby know that you are available to create a Wi-Fi Direct connection. This method call is not strictly needed, because other nodes eventually will detect your device as they periodically scan for nearby devices; however, sending this message will make them notice your device faster.

Insert metadata and resource information, on this node, using:

```
itrust_library.node.insert_resource_with_keywords(“r1”, {“k1”,
…});
```

This method call will insert some resource r1 with an array of keywords k1 (i.e., the metadata). The resource string can be simply text, but iTrust will automatically search your device to see if the resource string matches a file name. If the string is indeed a file name, then iTrust will send the file data instead of the string when another node retrieves a file. For example, suppose you are writing a camera app that takes a picture and you save it with the name P1.jpg and enter the string P1.jpg as the resource string. When another node attempts to retrieve the picture P1, iTrust will send the P1.jpg file itself and not the string name (which is what most users expect).

To send a query q1, use:

```
itrust_library.signal.send_query(“q1”);
```

The query q1 will be sent to a subset of nodes in the membership chosen at random. The query information, such as date sent, etc., will be tracked automatically by iTrust over Wi-Fi Direct in an internal SQLite database; your app does not need to handle multiple queries or multiple query matches.

The following iTrust over Wi-Fi Direct methods are self-explanatory and complete the API example:

```
itrust_library.signal.send_metadata_notice();
itrust_library.signal.request_metadata();
itrust_library.signal.get_all_resources();
```