

CIPHERMAIL EMAIL ENCRYPTION

---

# **Ciphermail for Android Quick Start Guide**

---



June 19, 2014, Rev: 5460



## 1 Introduction

This quick start guide helps you to setup Ciphermail for Android and explains how to use Ciphermail with the Android Gmail application.

---

**Note:** this quick start guide assumes that the reader does not have any prior knowledge of encryption and S/MIME. In order to get you going quickly, the software will create a certificate during the installation process and some of the default settings are setup to make it easier for end users to start encrypting email. “Self signed” means that the certificate is not validated by a Certificate Authority. We advise you to install a proper certificate after installation of Ciphermail. For more details, we suggest you read the *Ciphermail for Android reference guide*.

---

## 2 Install

Ciphermail for Android can be installed from the Android Market or from the Ciphermail website ([www.ciphermail.com](http://www.ciphermail.com)). On first use, a configuration wizard will be started which will guide you through the required configuration steps.

## 3 Configuration wizard

### 3.1 Step 1 - Introduction

The first page of the wizard shows some general information about the wizard (see figure 1).

### 3.2 Step 2 - Account

The account is the email address from which you will be sending email, i.e., the *from* address (see figure 2). Your Gmail address will be used by default.

### 3.3 Step 3 - SMTP settings

Ciphermail for Android requires an external SMTP server for sending email. The wizard automatically configures your SMTP settings if you are using Gmail. If you want to setup a different outgoing server, click the *Manual setup...* button (see figure 3).

#### 3.3.1 Step 3 - Gmail account

Sending email via the Gmail SMTP servers, requires the Gmail username and password (see figure 4).

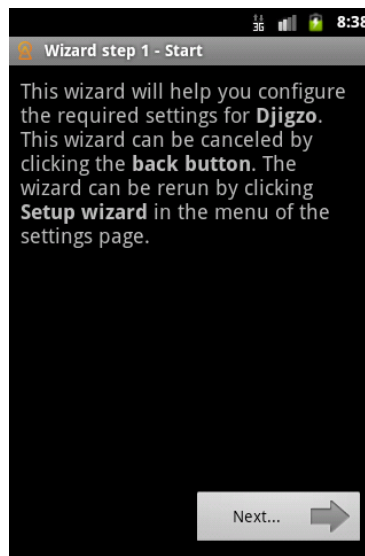


Figure 1: Wizard Step 1 - Introduction

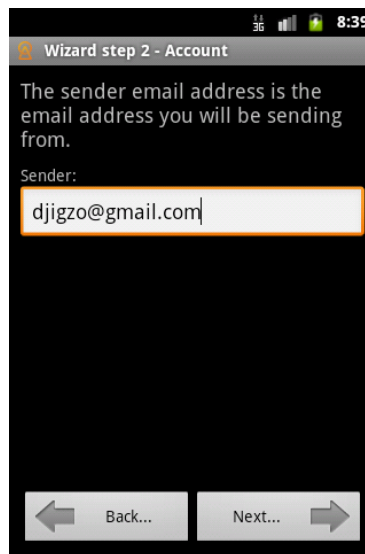


Figure 2: Wizard Step 2 - Account

**Note:** if you selected *Manual setup...* in the previous step, please refer to the *Ciphermail for Android reference guide* for instructions on how to manually configure the SMTP account.

### 3.4 Step 4 - Certificate

Ciphermail for Android supports encryption and digital signing using S/MIME. S/MIME is a standard encryption protocol for email, defined by the IETF. A

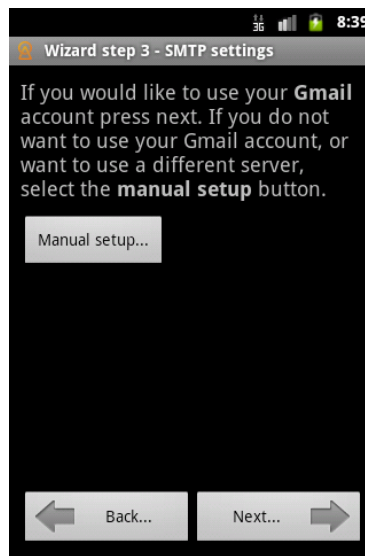


Figure 3: Wizard Step 3 - SMTP settings

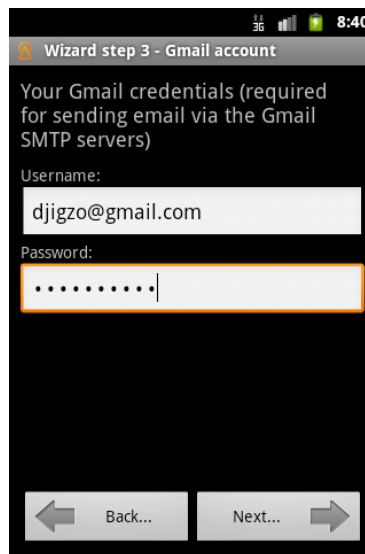


Figure 4: Wizard Step 3 - Gmail account

certificate and private key is required for S/MIME. The wizard will automatically generated a self-signed certificate for you (see figure 5). *self-signed* means that your certificate is not signed by a Certificate Authority (CA). We advise you to install a certificate from a CA.

**Note:** if you already have a certificate or would like to use a certificate issued by an external CA, you can skip this step. Please read the *Ciphermail for Android reference guide* for instructions on how to import an existing or newly

purchased certificate and its private key.

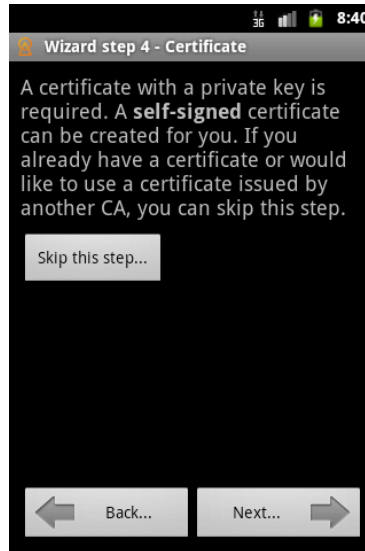


Figure 5: Wizard Step 4 - Certificate

#### 3.4.1 Step 4 - Generate certificate

A certificate requires an email address and the name of the owner (see figure 6). The certificate will be generated in the next step (see figure 7).

**Note:** depending on the speed of the Android device, generating a certificate and key may take some time.

#### 3.4.2 Step 4 - Key store password

Private keys for your certificates will be stored in a password protected *key store*. The first time the key store is accessed, you need to select a key store password (see figure 8).

**Note:** the system will prompt you for the key store password every time the key store is accessed. The password will be cached until the key store password times out. The key store timeout can be set in the key store settings. The default key store timeout is 60 seconds. You may find it convenient to set this to a higher value.

### 3.5 Step 5 - Finish

The wizard is now finished.

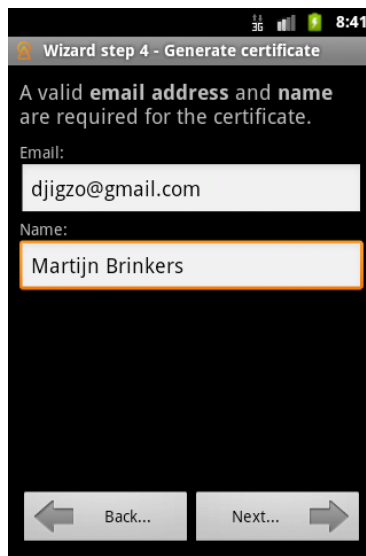


Figure 6: Wizard Step 4 - Generate certificate

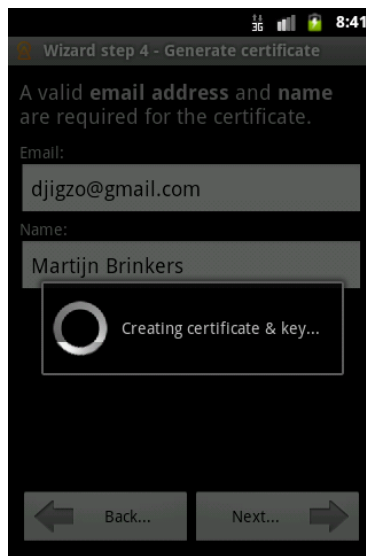


Figure 7: Wizard Step 4 - Generating certificate

**Note:** the wizard can be restarted by opening the settings screen and select the *Setup wizard* menu item.

## 4 Exchanging certificates

Before you can send an encrypted email to someone, you need to get their certificate, and they need to have yours.

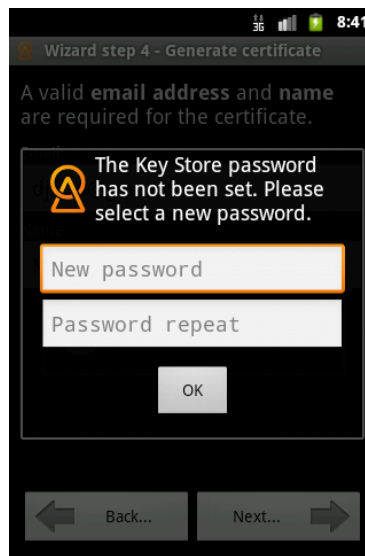


Figure 8: Wizard Step 4 - Key store password

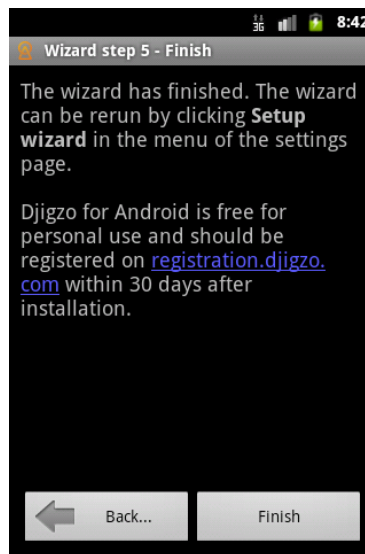


Figure 9: Wizard Step 5 - Finish

## 4.1 Sending your certificate

Your certificate can be sent to a recipient using the *Send My Certificate* option (see figure 10). The *Send My Certificate* page can be used to send your certificate to any recipient (see figure 11).

**Note:** an alternative method of exchanging certificates is by sending a digitally signed message to someone. The certificate that the message is signed



with, will be added to the digital signature of the message.

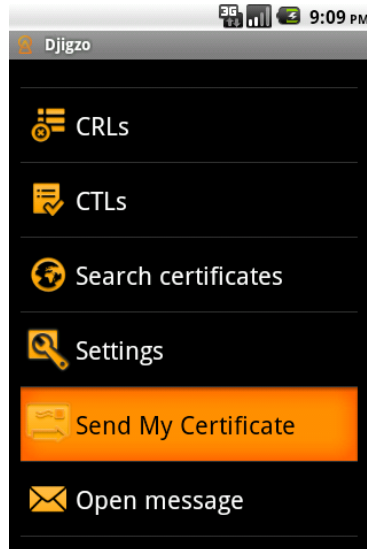


Figure 10: Send My Certificate Menu

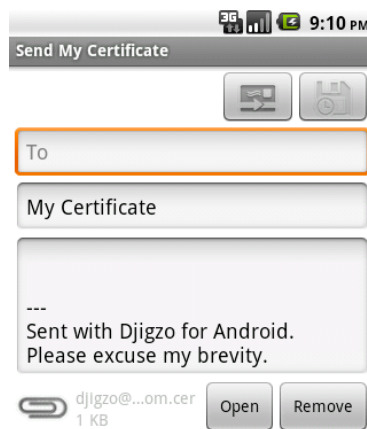


Figure 11: Send My Certificate

## 4.2 Receiving a certificate

Since you need the certificate of the recipient, you should ask all your recipients to send you their certificates (see figure 12). The certificate can be imported

into Ciphermail by previewing the attached certificate (see figure 13). By clicking the *Import certificate* button, the certificate will be imported.

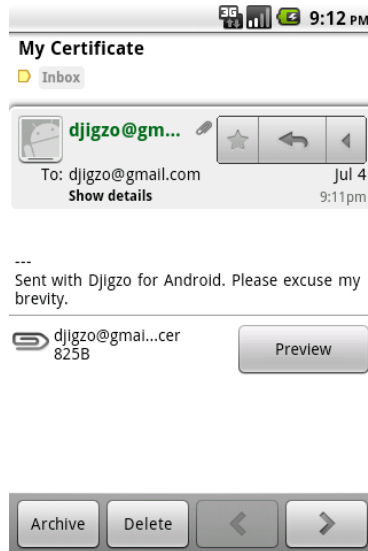


Figure 12: Receiving a Certificate

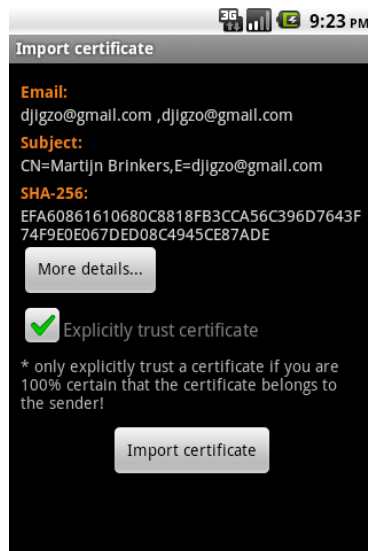


Figure 13: Importing a Certificate

### 4.3 Certificate trust

A recipients certificate is only used when the certificate is trusted. S/MIME is PKI based. In PKI, *trust* is based on a hierarchical model where trust is inher-

ited from the issuer of a certificate. To make full use of this hierarchical trust mode, you need to use certificates that are issued by trusted issuers (i.e., certificates issued by issuers from the root store). Since the generated certificate is self-signed, you need to explicitly trust the certificate in order to use it. By selecting the *Explicitly trust certificate* checkbox, the certificate will be trusted even if the certificate is not issued by a trusted root<sup>1</sup>.

**! Note:** only explicitly trust a certificate if you are 100% certain that the certificate belongs to the sender of the certificate!

## 5 Sending encrypted and signed email

An email can be encrypted, digitally signed and sent by clicking the *Compose message* menu item (see figure 14). On the compose message page allows you can set the recipients, add attachments, select whether the message should be signed and/or encrypted etc. (see figure 15).

**Note:** a message can only be encrypted for a recipient if a trusted certificate is available for the recipient.

## 6 Receiving encrypted email

An S/MIME encrypted email in Gmail will be shown as a normal email with an *smime.p7m* attachment (see figure 16). The *smime.p7m* attachment contains the encrypted message. The encrypted message can be opened by clicking the *Preview* button. The encrypted message will be opened by Ciphermail and will be decrypted (see figure 17). The example message from figure 17 is an HTML message which was encrypted and digitally signed. The digital signature was valid and the signing certificate was trusted. Using the main menu of the activity, you can reply to the message, forward the message etc.

**Note:** opening an encrypted message requires you to enter the key store password (if the password has timed-out).

## 7 Final note

Ciphermail contains a large number of user configurable settings. Some of these settings determine the level of security provided by Ciphermail for Android. As with all security products, there is a trade-off between user friendliness and security. We will briefly highlight some of the security related settings that are most important. For more details we suggest you read the *Ciphermail for Android reference guide*.

---

<sup>1</sup>By default *Explicitly trust certificate* is selected when the certificate is imported. The default setting can be changed in the general settings

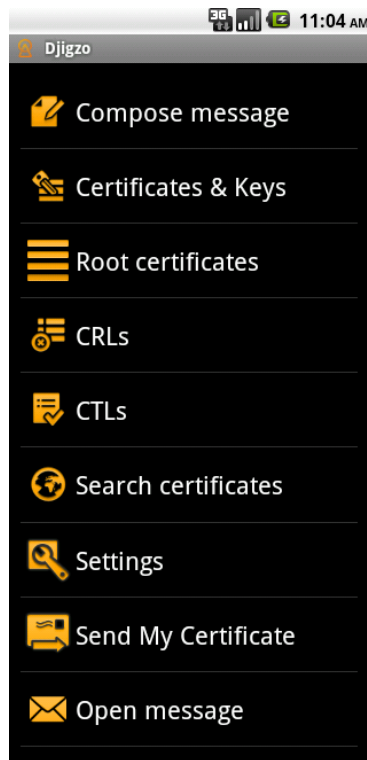


Figure 14: Main menu

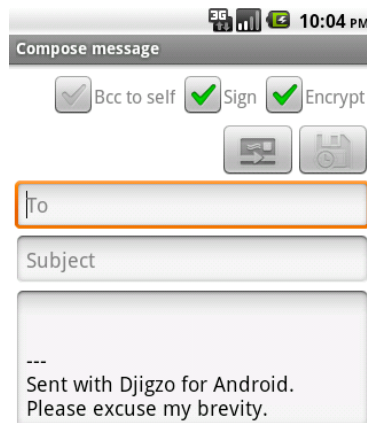


Figure 15: Compose message

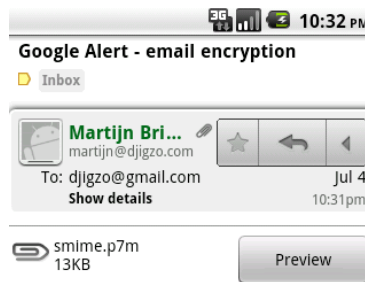


Figure 16: Encrypted email

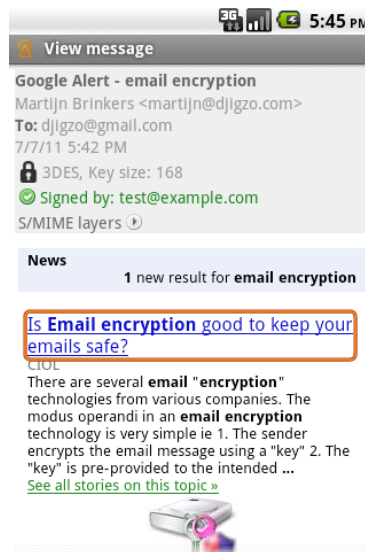


Figure 17: Decrypted email

## 7.1 Key store password

All the private keys are encrypted with the key store password<sup>2</sup>. The key store password should be long enough and should not be easily guessable.

<sup>2</sup>To be precise, the private keys are encrypted with a randomly generated key and the random key is encrypted with the key store password.

**Note:** to protect against a “brute force” attack, the private keys should ideally be stored in a hardware security device (for example a smartcard). Upcoming versions of Ciphermail for Android will support hardware security devices.

## 7.2 Password timeout

The key store password will be cached by default for 60 seconds. When the password has timed-out, you need to reenter the key store password when the key store is accessed. The password timeout can be set to a higher value.

## 7.3 Explicitly trust certificate

Ciphermail allows the use of self-signed end user certificates. Instead of importing these self-signed user certificates into the root store (since that would be insecure), the self-signed end user certificates can be explicitly trusted<sup>3</sup>.

When a single certificate is imported, the import activity can automatically trust the certificate (see figure 13). The certificate should only be explicitly trusted if the receiver knows with 100% certainty that the certificate belongs to the sender. Whether or not the *Explicitly trust certificate* is checked by default is determined by the setting *Trust certificate on import* in the *General settings*. To make it less likely that the user explicitly trusts the certificate on import, uncheck the *Trust certificate on import* setting.

## 7.4 Encryption and signing algorithm

By default, email is encrypted with AES128 and signed with SHA1. This is secure enough for all email. If a higher (or lower) level of security is required you can select different encryption and signing algorithms.

**Note:** not all email clients support all the available algorithms. For example, SHA256 (also known as SHA2) is only supported on newer versions of Windows.

---

<sup>3</sup>A certificate is explicitly trusted when the certificate is placed on the certificate trust list with “whitelisting” enabled.